

UVM项目实战部分5

路 桑

Agenda

1. 设计描述信息的标准化
2. 寄存器描述文件
3. UVM寄存器模型的自动化



设计描述信息的标准化

介绍

- IP-XACT是一种标准化的、格式化的设计信息描述标准（IEEE 1685-2014）。
- IP-XACT定义了一种标准化的方式去描述一个IP的关键信息，例如IP的用户可以从IP一致化信息中访问数据或者实现设计的自动化。
- 对于IP用户，当他们使用外部商业IP或者内部IP时，一般会得到大量从不同角度描述IP的文件，这包括有RTL代码、使用文档、仿真模型、综合约束。
- 但是不同提供商、不同类型的IP在这些文档上面，依然没有统一规范，而且信息还经常不充分。

设计描述信息的标准化 介绍

- IP-XACT没有试图定义哪些内容需要提供，也没有规范这些内容如何组织。
- IP-XACT试图去定义一个电子版的数据手册，用来归档不同种类的信息。
- IP-XACT标准最常用的场景包括：
 - 定义IP模型的关键信息，例如顶层端口名。
 - 提供入口用来指引交付的IP文档所包含的不同信息。
 - 用来对由IP-XACT建模的IP提供配置信息和互联信息。

设计描述信息的标准化

介绍

- IP-XACT语言规范本身也建立在XML格式之上，所以IP-XACT标准本身阅读起来很吃力，因为其标准都是对机器友好的（machine readable）。
- 因此，学习IP-XACT的重点在于如何应用它，理解这种规范化设计数据文件的优点和用处。
- 与IP-XACT标准化设计信息类似的数据存储格式，还包括SystemRDL，XML，RALF，CSV，Excel等。
- 对于verifier的日常工作，我们需要面对的，可能是IP-XACT文件格式，也可能是上述某一种标准化数据的存储格式，我们需要理解不同的人利用这些数据可以实现和处理什么。

设计描述信息的标准化

IP-XACT基础

- IP-XACT基于XML格式，它可以提供不同对象的描述信息：
 - 组件（component），即IP层次信息，包含IP的参数、寄存器、端口、接口（端口列表），这些信息层次较为抽象和宽泛。
 - 设计（design），包含层次化的设计信息，例如地址表、物理结构。组件和设计信息可以共同描述层次化IP的集成接口和内部结构。
 - 设计配置（design configuration），指的是针对具体的IP应用常见，基于设计IP-XACT文件，做出对应的设计配置，用来进一步决定设计实例的结构。一个设计文件可以对应多个设计配置文件。
 - 总线定义（bus definition）和抽象级定义（abstraction definition）用来描述硬件的通信协议。总线定义用来指定具体的总线类型，而抽象级定义用来指定通信传输时的层级。

设计描述信息的标准化

IP-XACT基础

- IP-XACT基于XML格式，它可以提供不同对象的描述信息：
 - 生成器流程（generator chain）用来描述IP-XACT得以转化的流程，例如哪一种工具或者脚本，哪些参数，可以最终利用IP-XACT文件生成标准化文件。
 - 目录（catalog）是用来管理多个IP-XACT文件，例如这些文件的链接（URL）和它们的说明。

设计描述信息的标准化

IP-XACT基础

- 由此可以看出IP-XACT格式能够实现的，不只是我们接下来会学习的寄存器信息，它还包括设计的：
 - 层次信息
 - 总线信息
 - 互联信息
 - 配置信息
 - 地址信息
 - 集成信息
 - ...
- 理论上，通过IP-XACT管理的IP，可以实现开发、交付、集成和维护的标准化。

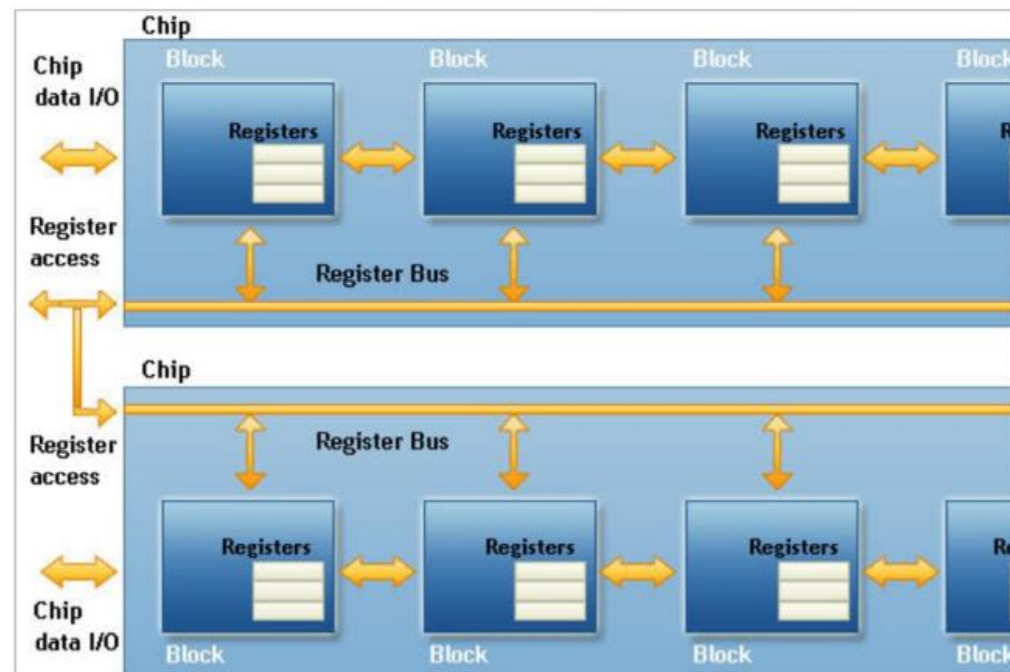
Agenda

1. 设计描述信息的标准化
2. 寄存器描述文件
3. UVM寄存器模型的自动化

寄存器描述文件 介绍

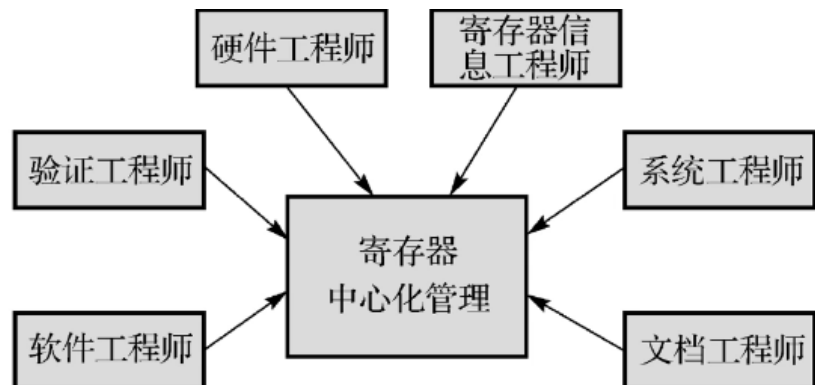
- 目前的数字设计可配置化程度很高，这满足了设计的灵活性和成本的降低。
- 同时，配置化数字设计带来了成百上千的寄存器，这些寄存器在数字化设计流程中扮演着关键的作用，因为它们都是硬件和软件之间的接口。
- 如同MCDF的寄存器模块，配置寄存器和状态寄存器构成了硬件的编程接口。
- 对寄存器信息的细致安全管理可以提高设计交付的质量。

寄存器描述文件 介绍



- 在SoC系统中，主要由处理器、功能模块、I/O模块、总线和互连网络构成。
- 多数的模块均可以由外部或者内部完成寄存器访问。
- 寄存器访问可以归结为总线和寄存器模块。
- 总线的标准化，和寄存器信息的标准化使得在寄存器信息标准化管理和自动化实现变为了可能。

寄存器描述文件 中心化管理

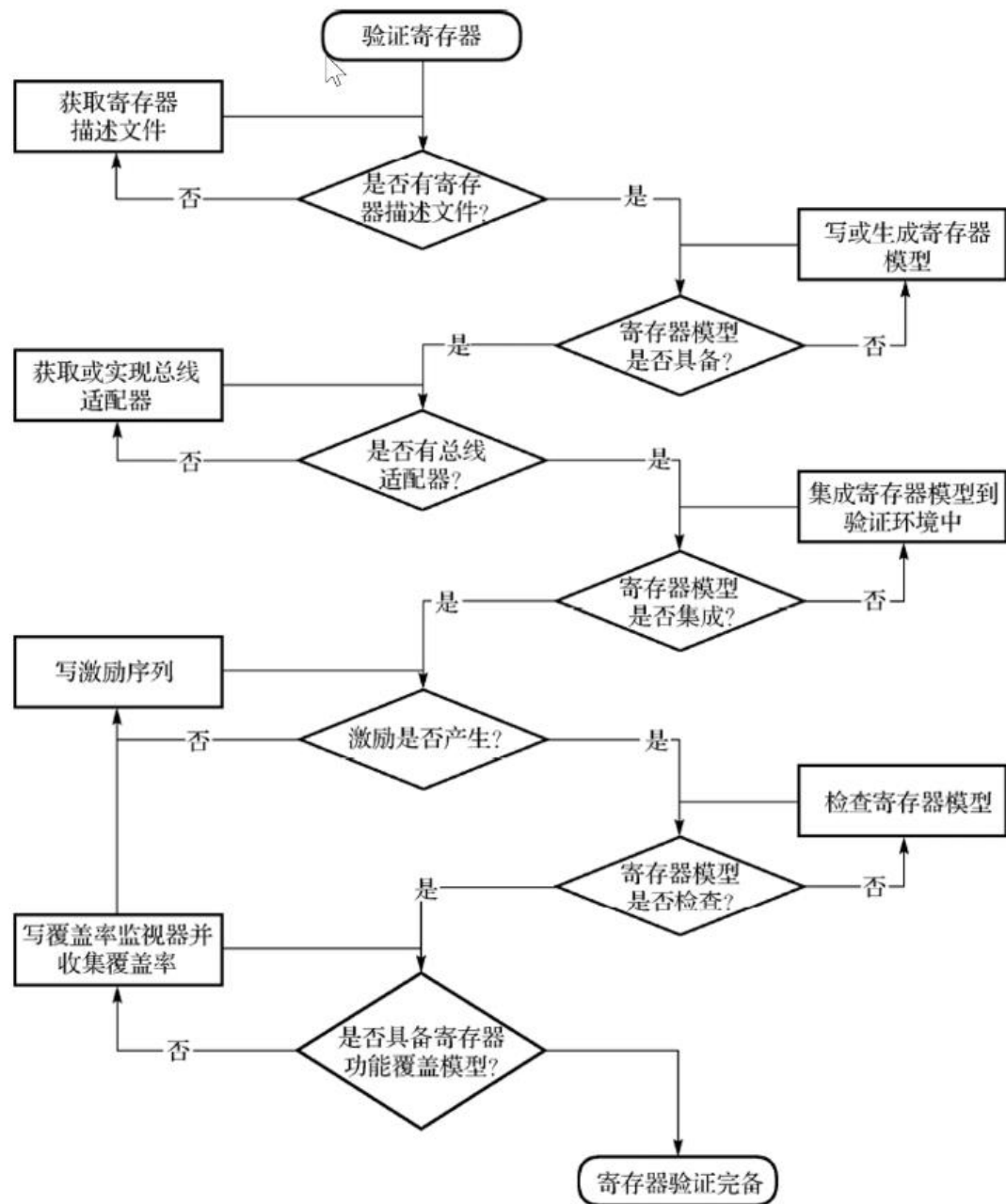


- 在UVM入门和进阶培训模块，关于UVM寄存器模型的介绍中，我们已经了解到，在系统开发过程中，不同角色的工程师都可以利用中心化管理的寄存器信息实现其流程：
 - 系统工程师会撰写并维护寄存器描述文件，而后归置到中心化存储路径供其他工程师开发使用。
 - 硬件工程师会利用寄存器描述文件生成寄存器硬件模块（包含各个寄存器的硬件实现和总线访问模块）。
 - 验证工程师会利用寄存器描述文件来生成UVM寄存器模型，以供验证过程中的激励使用、寄存器测试和功能覆盖率收集。
 - 软件工程师会利用该文件生成用于软件开发的寄存器配置的头文件（header file），从而提高软件开发的可维护性。
 - 寄存器描述文件也可以用来生成文档，实现更好的可读性。

寄存器描述文件

寄存器模型生命周期

- 系统工程师提供寄存器描述文件。
- verifier生成寄存器模型。
- VIP开发人员提供总线适配器（adapter）。
- TB构建人员集成寄存器模型。
- verifier也需要完成寄存器模型检查和覆盖率收集。



寄存器描述文件

案例学习 (IDesignSpec)

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.

SoC Enterprise™ (SoCE)

- Enterprise SoC and IP Creation and Management (in Beta)

ARV-Sim™

ARV-Formal™

IDesignSpec™

IDSBatch / IDSWord / IDSEExcel / IDSCal

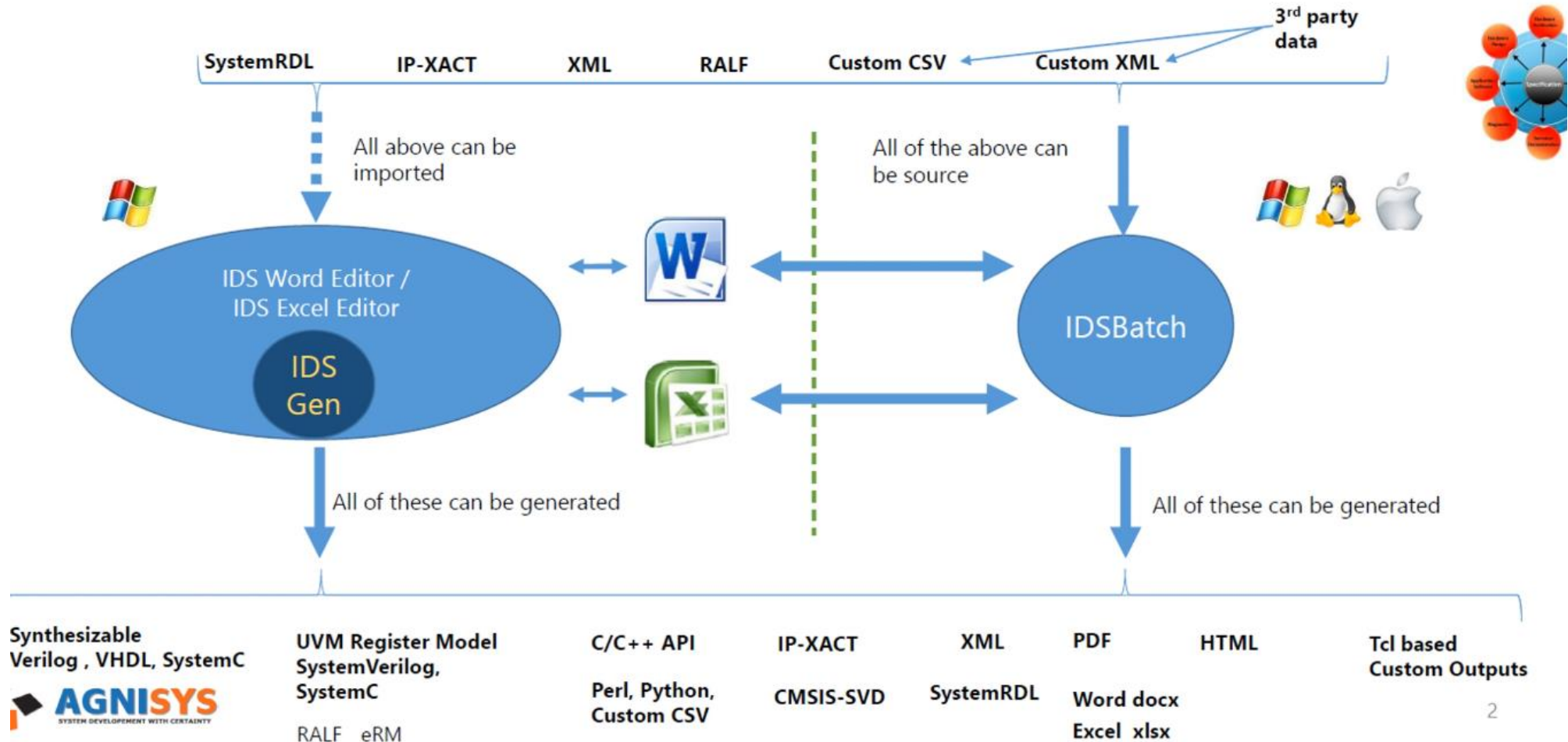
ISequenceSpec™

DVinsight™

SoC Enterprise™ - (Beta)

寄存器描述文件

案例学习 (IDesignSpec)



寄存器描述文件

案例学习 (IDesignSpec)

Make changes to the Specification and have the change automatically permeate to all views



1. Keep existing work flows.
2. Enable self-checking specs.
3. Customizable outputs
4. Extensible and flexible solution
5. Should be simple - no PhD required
6. Additionally ...
 - Interactive Vs Batch
 - Windows Vs Linux
 - Standards support
 - Import legacy data
 - Support complex structures

寄存器描述文件

寄存器信息编辑

- 一些商业寄存器编辑器大致会包含以下内容：
 - 寄存器块 (register block)
 - 寄存器 (register)
 - 寄存器域 (register filed)

IDS_blk		IDS_blk		Block 	0x00001
offset	01	External		Size	

IDS_reg_A		IDS_reg_A		Reg. 	0x00011
offset	'h10	External			

Sample Register of Block IDS blk

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
bits		name				s/w		h/w		default			Description																		
31:0		Fld_1				Rw		Ro		'hA			Field for the IDS_reg_A register with reset value equal to 'hA																		

寄存器描述文件

寄存器信息编辑

- 不少公司可能会自己内部开发寄存器信息编辑器（可视化的或者文本处理的），然后再通过脚本、模板来实现设计自动化和验证自动化。
- 对于verifier，我们将模拟实际场景，采用Excel这种标准化的数据存储方式，结合Python脚本来实现UVM寄存器模型自动化。
- 我们本次实战的目的在于认识，设计信息标准化的重要性，以及利用标准化设计信息可以实现的设计和验证过程加速。

Agenda

1. 设计描述信息的标准化
2. 寄存器描述文件
3. UVM寄存器模型的自动化

UVM寄存器模型的自动化

- 随堂练习：
 - 理解路桑给的寄存器信息描述文件（excel）的内容。
 - 掌握如何使用Python脚本来线下或者在线生成UVM寄存器模型。
 - 根据MCDF寄存器描述的更新来相应修改寄存器文件（excel）。
 - 通过Python脚本来自动生成寄存器模型SV文件。
 - 检查新生成的寄存器文件，与MCDF的寄存器描述进行参照比对，确保信息的一致性。
 - 用现有的寄存器模型SV文件来替换原有的寄存器模型SV文件，完成编译，确保新生成的为寄存器文件的语法正确。