A Refresher on SystemRDL (Register Description Language)

First let's start with the basics: Registers are how SOC functionality is accessed, such as USB controllers, network cards, graphics, hard drive control, etc. Whether or not they are included in the CPU package or as a separate hardware module, these services are accessed through registers. Registers are used for configuration, status query and ongoing control and without them, devices can't do much at all.

SystemRDL is a Register Description Language for the design and delivery of intellectual property (IP) products used in designs. SystemRDL semantics supports the entire life-cycle of registers from specification, model generation, and design verification to maintenance and documentation. Registers are not just limited to traditional configuration registers, but can also refer to register arrays and memories.

The SystemRDL specification was released in May 2009 by the The SPIRIT Consortium. The SystemRDL 1.0 standard was transferred to Accellera upon its merger with The SPIRIT Consortium. The intent of this standard is to define SystemRDL accurately.

SystemRDL is designed to increase productivity, quality, and reuse during the design and development of complex digital systems. It can be used to share IP within and between groups, companies, and consortiums. This is accomplished by specifying a single source for the register description from which all views can be automatically generated, which ensures consistency between multiple views. A view is any output generated from the SystemRDL description, e.g., RTL code or documentation.

Unlike most standards that target a specific audience, SystemRDL is important to everyone. Operating Systems need registers to query CPU status, manage memory, and to display information on the screen. Register implementation needs to be in sync with device drivers, documentation, validation, manufacturing test, and more.

While the current SystemRDL standard is good, it has some areas that need improvement. The SystemRDL Working Group is currently addressing:

- Its coordination with related standards such as UVM and IP-XACT to ensure that the register field access types and side effects are compatible and consistent
- Complexity management: Currently there's a single dimension instantiation, but products have multiple cores, multiple graphics, etc.; Properties are currently name/value pair only – structs are needed for better organization; to reduce errors due to misspelled property values (currently strings), enumerations are being extended for use as property values
- Configurability: Component parameterization, property assignment arithmetic and content deprecation (the ability to selectively turn off subcomponents), thereby enabling a single set of RDL files to represent multiple SKUs/steppings/bins, etc.
- Standardization of constraints, memories, HDL paths

The need for SystemRDL continues to grow. Greater integration of SOCs has resulted in an explosion of registers. Many CPU's have 100,000+ registers. Companies are beginning to understand that Excel is not sufficient and hand-coded RTL is error-prone. More and more IP is shared and licensed between companies and validation, device drivers, documentation, etc. all need precise information. SystemRDL is the only standard for registers that supports the needs of both design capture and communication of register specs to downstream consumers.