

Comparison of CSRCompiler Input Formats

Executive Summary

The CSRCompiler™ system supports several input formats for register and address map information. The formats are: the CSRSpec™ language, IEEE 1685 IP-XACT, CMSIS-SVD, spreadsheet CSV/XLS and Spirit SystemRDL. The CSRCompiler is capable of cross compiling from one format to another and is able to link most formats to the others.

The CSRSpec language is not just a superset of all industry standard formats, it is also the next generation address map authoring format since it provides the entire CSRCompiler feature set to the author. We recommend this format.

The IEEE 1685 IP-XACT XML format is meant for exchange of third-party IP core information. This format is not appropriate for authoring the address map source. Although the format does have information about the layout and software access of the address map registers, it does not have implementation information and is not appropriate for RTL generation.

The CMSIS-SVD format is meant for exchange of register address information with software development environments. This format is not appropriate for authoring the address map source. Like IP-XACT it has layout information for software development. However, it also does not have implementation information and is not appropriate for RTL generation.

The spreadsheet format is an authoring format for small designs with limited features. This format becomes unmanageable very quickly as the design size grows. Also, many features supported by the other formats cannot be represented in this format. We recommend this format only for smaller designs with limited features.

SystemRDL is an old lapsed standard that has fallen behind the current capabilities of the mainstream active standards. This format is missing basic addressmap features like a memory instance. We do not recommend this format for new designs. This input format is supported for import of legacy design information.

Input Format Considerations

First, we recommend CSRSpec as the input format. This format will scale with your design size and supports all the possible features and there is the possibility of timely enhancements to get your design completed. Some of the formats are industry standards which only are enhanced once every five years. Some of the formats are very limited in their scope. The XML formats are meant for tool use only and would not be appropriate as an authoring format. Some of the formats become unmanageable quickly as the design grows in size or features. However, there are reasons to use the formats. You may use several input formats in the same design, depending on the source of the information.

The IP-XACT XML format is only recommended for third-party IP core address map register information exchange between tools. This XML format is very complex is not recommended for authoring the address map design. This standard is only meant for documentation of the hardware and software interfaces in a design, and, this format is missing implementation information. There are several versions of the IP-XACT format, and, only the latest IP-XACT has the minimum set of address map register features for use in a current environment. Unfortunately, this standard allows vendor extensions to be added to the information; however, these extensions are not standard, and the information contained in vendor extension is not useable and does not transfer between tools. It seems many of the files are created by hand or by bad tools. Part of the reason that some of the files are bad is that the standard does not actually support all the features of the design.

The CMSIS-SVD format is for exchange of address map register information with software development environments. This format only has the software facing information. The details for RTL generation are not present in SVD. This format is not recommended for authoring the design. This format should only be used for input if the information is not available in any other form.

The spreadsheet format is common in the industry as a simple way to document the register address map. However, as the size of the address map grows the number of objects and likewise the number of rows becomes unmanageable. Also, for address maps which contain fields with a rich set of behaviors the number of properties and likewise the number of columns becomes too large to view easily. The feature set is restricted when compared to other authoring formats. We recommend that spreadsheets only be used from smaller and simpler designs or if there is an existing compatible spreadsheet. As the design increases in size or complexity we recommend that you use the CSRCompiler to translate to CSRSpec.

The SystemRDL format should only be used to import legacy design information already in this format. Due to the age of this standard several important features in common usage today are not present. This format does not support memory objects that would be useful in verification environments like UVM. Since this format does not handle expressions, it has a limited capability to include “C” or Verilog header files for text macros. Although it is not recommended for new designs, many common legacy designs include overlapping read-only/write-only registers or fields, and, it is not possible to represent these registers or fields in SystemRDL. Although SystemRDL is meant for RTL generation, the feature set is barely adequate for simple designs.

The CSRCompiler distribution contains a conversion utility to convert from the RALF file format to CSRSpec. This utility is meant to support legacy designs where the only available register information is in the RALF format.

Table 1 Comparison of Input Formats

Format	CSRSpec	IEEE IP-XACT	CMSIS-SVD	SystemRDL	Spreadsheet
Recommended Usage	All designs large and small	Third-party IP core import	Interchange with software development environments	Import of legacy designs	Small designs with a limited feature set.
Address map Feature Count	200	25	13	69	80
Parameterized Templates	YES	NO	NO	NO	NO
Wide Registers	YES	NO	NO	NO	YES
Atomic register access	YES	NO	NO	NO	NO
Memory object	YES	YES	Limited	NO	YES
Virtual structures in memories	YES	NO	NO	NO	NO
Include of C or Verilog macros	YES	NO	NO	Limited	NO
Bus Protocol specification	YES	YES	NO	NO	YES

Slave interface port name control	YES	YES	NO	NO	NO
Multi-dimensional arrays	YES	YES	YES	NO	YES
Field Arrays	YES	NO	NO	NO	YES
Supports overlapping read-only/write-only fields	YES	YES	Unspecified	NO	YES
Overlapping read-only/write-only registers	YES	YES	Unspecified	NO	YES
Register and field cross references	YES	NO	NO	YES	YES
Register Alias	YES	NO	NO	YES	YES
Writedata field type	YES	NO	NO	NO	YES
Counter field type	YES	NO	NO	YES	YES
Interrupt field type	YES	NO	NO	YES	YES
Concatenation of cross-references	YES	NO	NO	YES	NO
Verification Backdoor path	YES	NO	NO	NO	YES
C volatile customization	YES	NO	NO	NO	NO
Documentation Privacy/Secrecy control	YES	NO	NO	NO	NO
Structured titles and descriptions	YES	NO	NO	YES	NO
Clock domain information	YES	NO	NO	NO	YES
Synchronizer support	YES	NO	NO	NO	YES

Semifore, Inc.

1000 Elwell Court, Suite 150B, Palo Alto CA, 94303

(650) 960 0200

sales@semifore.com www.semifore.com

Semifore, CSRCompiler, CSRSpec, and the Semifore logo are trademarks of Semifore, Inc. All other trademarks are the property of their respective owners.