PeakRDL-Markdown

Marek Pikuła

Nov 23, 2022

CONTENTS

This package implements a Markdown exporter for the PeakRDL toolchain.

CHAPTER

ONE

INSTALLING

Install from PyPi using pip:

```
python3 -m pip install peakrdl-markdown
```

If you want to use official PeakRDL CLI you can install with `cli` extra:

```
python3 -m pip install peakrdl-markdown[cli]
```

CHAPTER

TWO

QUICK START

2.1 Exporting to Python interface

The module integrates with PeakRDL CLI interface (via optional extra `cli`):

```
peakrdl markdown input_file.rdl -o output_file.md
```

CHAPTER

THREE

LINKS

- Source repository
- Release Notes
- Issue tracker
- PyPi

3.1 Using with Sphinx

This PeakRDL extension can be used to include the SystemRDL description in Sphinx documentation. For now, there is no direct support for reStructuredText (will be added in future versions), but you can use `m2r2` Sphinx extension to import Markdown files. You can find a complete guide how to install and enable `m2r2` extension on the project's website.

3.1.1 Example

As an example, we can use the following SystemRDL source:

```
addrmap some_register_map {

  name = "RDL Example Registers";
  desc = "This address map contains some example registers to show
          how RDL can be utilized in various situations.";

  reg {
    name = "This chip part number and revision #";
    desc = "This register cotains the part # and revision # for XYZ ASIC";

    field {
      hw   = w;
      sw   = r;
      desc = "This field represents the chips part number";
    } part_num[31:4] = 28'h12_34_56_7;

    field {
      hw   = na;
      sw   = r;
      desc = "This field represents the chips revision number";
```

(continues on next page)

(continued from previous page)

```
    } rev_num[3:0] = 4'b00_01;
  } chip_id_reg @ 0x0;

  reg {
    name = "Enable register";
    desc = "Toggle the peripheral enable on write";

    field {
      name = "Enable toggle";
      hw = r;
      sw = w;
      onwrite = wot;
    } enable[0:0];
  } enable @ 0x4;
};
```

The generated Markdown file can be included with the following Sphinx statement:

```
.. mdinclude:: minimal_example.md
```

It generates the following output:

3.1.2 some_register_map

- Absolute Address: 0x0
- Base Offset: 0x0
- Size: 0x8

Offset	Identifier	Name
0x0	chip_id_reg	This chip part number and revision #
0x4	enable	Enable register

chip_id_reg

- Absolute Address: 0x0
- Base Offset: 0x0
- Size: 0x4

Bits	Identifier	Access	Reset	Name
3:0	rev_num	r	0x1	—
31:4	part_num	r	0x1234567	—

rev_num

part_num

enable

- Absolute Address: 0x4
- Base Offset: 0x4
- Size: 0x4

Bits	Identifier	Access	Reset	Name
0	enable	w, wot	—	Enable toggle