low-cbf-pss-interface Documentation

CSIRO

Aug 14, 2022

Contents

Shared communications interface between Low CBF and PSS.

The source of truth is the C++ code, used directly by PSS. CBF will use VHDL derived from this file via a Python script.

CHAPTER 1

Conversion Process

The python script 'translate.py' does the C++ to VHDL conversion. It requires the *castxml* command-line utility to be installed, and the pip package *pygccxml*.

The most up-to-date information regarding command line arguments is available by running the program with the *-h* argument.

```
$ python3 translate.py -h
usage: translate.py [-h] [--namespace [NAMESPACE]] infile [infile ...] outfile

C++ struct to VHDL translator

positional arguments:
  infile
  outfile

optional arguments:
  -h, --help            show this help message and exit
  --namespace [NAMESPACE]
```

- One or more **infile** arguments specify the C++ source files to read
- One VHDL output file is generated, as specified by **outfile** (use '-' to output to standard out)
- An optional **namespace** can be provided - if not supplied, the global namespace is used. This may be problematic if your source files include libraries containing structs.

Note that structs with names beginning with '__' are ignored. This was done to avoid producing VHDL for some library internal structures.

Bit widths of C++ data types are not always deduced by the parser (reason unknown). If pygccxml does not provide a byte width, we read numbers from the type name and assume they are bits.

I suggest restricting the C++ source to the (u)int<n>_t types from cstdint - this provides clarity for C++ readers too.

CHAPTER 2

Indices and tables

- genindex
- modindex
- search