

vdt specification

[Document Subtitle]

RyanHunter

[personal]

1. overview

this document describes the whole strategy for developing vdt-v1.x version tools.

TBD, omitting some redundant descriptions

1. usage&features

This chapter will list all possible usage/issues or features this tool are going to solve, it's kind of a very begining state to collecting requirements and provide the features, And because of the lack of experience, this chapter may continue updating.

* 1. designs

A design here refers to a module in verilog, we can call 'design' with a bunch of code to declare a design module.

* 1. signals

There're two ways to publish signals to RTL code, one is explicitly declared by a global or module perspective signal method, another way is to implicitly infered by the tool.

As for explicit declaration, users need to call with 'signal' method, with args such as type/name/width, however, it might be tedious to declare many signals by calling that method, so we need a way to facilitating the signal declaration.

* + 1. signal pool

Signals can be declared within a signal pool, which is like a namespace, by importing that signal pool, a design module can then chose which of those signals to use.

Signal pool are much like to be used by declaring ports of a design.

* + 1. flexible signal operations

There might be two signals that have the same name, but different width or type. So we can first declare a signal in global scope, and then declared within a design with a succeeding method. (TBD)

* 1. components

a component is a bunch of logical operations that can be reused (by instantce mechanism) in different modules, but it actually will not directly reflect on the target RTL file. It's simply a reusable block. Users can easily declare a new component similarily as declaring a design.

* + 1. declaration

By calling component literaly will trigger a declare action to create and register a new component in the global scope, every design can instantiate that component directly with some extra instantiating operations.

* 1. logicals

The essence operations of a integrated circuit is kind of a logical operation, so logicals are at the bottom layer of those component/design ... layers.

1. implementation&strategies

This chapter gives strategies and implementations for vdt development.

TBD