

Clean up your EDA flows with `tclint`

Noah Moroze

me@noahmoroze.com

Open Source EDA BoF @ DAC '24 • June 25, 2024

Work presented is my own.

tclint checks Tcl scripts for issues

place.tcl

```
1 if {$optlevel > 1} {  
2   place_design -effort high  
3 } elseif {optlevel > 0} {  
4 place_design -effort low  
5 }  
6  
7 check_place -out "place.rpt"  
8 save_design "design.def"
```

```
$ tclint place.tcl
```

```
place.tcl:3:9: syntax error: invalid  
bareword in expression: optlevel
```

```
place.tcl:4:1: expected indent of 2 spaces,  
got 0 [indent]
```

```
place.tcl:7:1: invalid option '-out', did  
you mean '-file' [command-args]
```

Linting is SW best practice



Flake8



Verilator



ESLint



Clippy



Vet

Linting is particularly valuable for EDA

place.tcl

```
1 if {$optlevel > 1} {  
2   place_design -effort high  
3 } elseif {optlevel > 0} {  
4 place_design -effort low  
5 }  
6  
7 check_place -out "place.rpt"  
8 save_design "design.def"
```

Bugs hide in branches

Runtime crashes can waste lots of time

Linting is particularly valuable for open source

The screenshot shows a GitHub pull request titled "Well-intentioned update #1". The pull request is open and shows that the user "nmoroze" wants to merge 1 commit into the "main" branch from the "my-patch" branch. The pull request has 0 conversations, 1 commit, 1 check, and 1 file changed.

A comment from "nmoroze" (the owner) says: "I didn't add tests but I swear this PR works on my machine." Below the comment is a commit titled "Mess up the Tcl code" with a red 'X' icon and the commit hash "649f3c2".

Below the comment, there is a message: "Add more commits by pushing to the `my-patch` branch on [nmoroze/cool-eda-project](#)."

At the bottom, there is a failed check section titled "All checks have failed" with 1 failing check. The failing check is "CI / Lint Tcl (pull_request)" which is required. There is a "Details" link next to it.

On the right side of the page, there are sections for "Reviewers" (No reviews), "Assignees" (No one assigned), "Labels" (None yet), and "Projects" (None yet).

tclint is being used in CI

- [siliconcompiler/siliconcompiler](https://github.com/siliconcompiler/siliconcompiler)
- [The-OpenROAD-Project/OpenROAD](https://github.com/The-OpenROAD-Project/OpenROAD)
- [foss-for-synopsys-dwc-arc-processors/toolchain](https://github.com/foss-for-synopsys-dwc-arc-processors/toolchain)



SiliconCompiler



The OpenROAD Project

tclint has caught bugs in the wild

```
4 tcl/memory.tcl
@@ -66,10 +66,10 @@ proc iswithin { ADDRESS BASE LEN } {
66 proc address_info { ADDRESS } {
67
68     foreach WHERE { FLASH RAM MMREGS XMEM UNKNOWN } {
69 -         if { info exists $WHERE } {
70             set lmt [set N_[set WHERE]]
71             for { set region 0 } { $region < $lmt } { incr region } {
72 -                 if { iswithin $ADDRESS $WHERE($region,BASE)
73                     $WHERE($region,LEN) } {
74
75             }
76         }
77     }
78 }

siliconcompiler/tools/openroad/scripts/sc_write_images.tcl
@@ -141,7 +141,7 @@ proc sc_image_routing_congestion {} {
141     sc_image_heatmap "Routing Congestion" \
142         "Routing" \
143         "routing_congestion.png" \
144 -         "routing congestion"
145     0
146 }
147 }
```

Command plugins check tool-specific scripts

- Currently supports OpenROAD
- Dynamically generated by reading help strings

```
$ tclint-plugins make-spec openroad -o or-commands.json  
$ tclint or_scripts/ --commands or-commands.json
```


There are many features on the roadmap...

- Command plugins
- Auto-format
- Parser library

... and you can shape its path!

Download

```
$ pip install tclint
```

Run

```
$ cd my-eda-flow/  
$ tclint .
```

Give feedback!

Scan the QR code to see the Github repo
Give it a star and/or open an issue!

github.com/nmoroze/tclint

