OPENDB::
TCL INTEGRATION TUTORIAL

OPENROAD

BY DIMITRIS FOTAKIS

PURPOSE

- Core EDA Tools use NP-Complete Algorithms with heuristics and Options
- End Users routinely runs tools with different Options for every new chip
- Tools are monolithic with their own APIs and program arguments
- Tcl is a Full Script language that can create a uniform programming environment for running different tools/algorithms and flows

OPENDB TCL SCRIPT EXAMPLE

```
# Initiate Run Time Stat
    db rlog
# Read Lef/Def
        db load_lef -file ../data/lef
        db load_def -file ../data/def
        db write –file before.place.db
# Run Placer
        set option1= “greedy”
        RPL p
        p init_place
        p place –intensive –algorithm $option1
        p show_stats
#save intermediate result
        p write_db
        db write –file placed.$ption1.db
```

OPENDB PROGRAMMING MODEL

- OpenDB builds automatically the TCl to C++ Interface classes to make it easy for the programmer
- For every module there are 3 areas for the C++ programmer to manage:
  - Tcl Command Definition: <Name> and options for Tcl Command
  - Tcl Command Class: C++ Class for the Tcl Command that includes APIs to the "main" module classes
  - Module Classes: C++ main Classes for the module that do NOT include any Tcl APIs

OPENDB BASH SCRIPT FOR A NEW MODULE

- makeNewModule.bash <MODULE> <MOD>
  - Creates a directory called MODULE under src of the OpenDB repository
    - MODULE.cpp – main class of the module
    - MODULE.h – main header
    - Makefile
  - Creates a directory MODULE/MOD_tcl
    - MOD – class of the Tcl interface
    - MOD.h – header of the Tcl interface class
    - MOD.ti – End User Tcl Command name and options
  - Updates src/Makefile, src/main/Makefile and src/main/modules.cpp
    - To make compilation and linking automatic

STEP-BY-STEP EXAMPLE

- Create a new module MyModule with Tcl Object Name MMD
  - cd src
  - ./BUILD/Templates/makeNewModule.bash MyModule MMD
  - make clean
  - make
  - make install
  - Edit bash stript BUILD/set_ade_env.bash
  - cd to any test dir
  - Run src/BUILD/set_ade_env.bash
  - ade – will create a prompt
  - MMD m
  - m – will list all tcl commands
  - m test_cmd – says hello
  - m test_cmd - : will give usage of the cmd
  - m test_cmd –hello Tom – says hello Tom

STEP-BY-STEP EXAMPLE - CONTINUED

- Add Integer Option on Command test_option of MyModule
  - Edit src/MyModule/MMD_tcl/MMD.ti
  - ZZAt the end of the line “in string msg …” add ,
    - and a new line : in int num = 0 [usage = "test int num with default 0" ;]
  - Edit src/MyModule/MMD_tcl/MMD.cpp
    - Add in_args->num() as a second argument in function call: _main->test_option
  - Edit src/MyModule/MyModule.h
    - Add int num as a second argument in function call: test_option
  - Edit src/MyModule/MyModule.cpp
    - Add int num as a second argument in function: test_option
    - Add additional code to print the value of num
  - make clean
  - make
  - make install
  - cd to any test dir
  - ade
  - MMD m
  - m test_option –msg “my msg” –num 10