Yet Another Synthesis Flow for RTL Compiler

Octavian Petre

NXP Semiconductors

May 21, 2014

Outline

Outline

Why Another RTL Compiler Synthesis Flow?
Less Error Prone (1)

- **Increasing Complexity**: RC scripts tend to grow with new projects and methodologies.
  I want to avoid long spaghetti code.

Why Another RTL Compiler Synthesis Flow?

Less Error Prone (1)

- **Increasing Complexity**: RC scripts tend to grow with new projects and methodologies.
  I want to avoid long spaghetti code.
- **Recommended Flow**: Cadence recommends certain steps in a certain order.
  I want to check them easily.

Why Another RTL Compiler Synthesis Flow?

Less Error Prone (1)

- **Increasing Complexity**: RC scripts tend to grow with new projects and methodologies.
  I want to avoid long spaghetti code.
- **Recommended Flow**: Cadence recommends certain steps in a certain order.
  I want to check them easily.
- **Missing Separation**: Customized steps/code are mangled with standard ones.
  I want to focus on the customized ones.

Why Another RTL Compiler Synthesis Flow?
Less Error Prone (1)

- **Increasing Complexity**: RC scripts tend to grow with new projects and methodologies.
  I want to avoid long spaghetti code.
- **Recommended Flow**: Cadence recommends certain steps in a certain order.
  I want to check them easily.
- **Missing Separation**: Customized steps/code are mangled with standard ones.
  I want to focus on the customized ones.
- **(Re)Set of Variables**: Variables (re)set in several places in different directory trees.
  I want to avoid reading all scripts

Why Another RTL Compiler Synthesis Flow?

Less Error Prone (2)

- **Missing Collaboration**: Fixed issues are not always picked up by new designs.
  I want to get all fixes fast.

Why Another RTL Compiler Synthesis Flow?
Less Error Prone (2)

- **Missing Collaboration**: Fixed issues are not always picked up by new designs.
  I want to get all fixes fast.
- **Insufficient Standardization**:
  - Flow steps.
    I want clear flow steps

Why Another RTL Compiler Synthesis Flow?
Less Error Prone (2)

- **Missing Collaboration**: Fixed issues are not always picked up by new designs.
  I want to get all fixes fast.
- **Insufficient Standardization**:
  - Flow steps.
    I want clear flow steps
  - Few, no, or difficult to understand procedures.
    I want clear API for procedures.

Why Another RTL Compiler Synthesis Flow?
Less Error Prone (2)

- **Missing Collaboration**: Fixed issues are not always picked up by new designs.
  I want to get all fixes fast.
- **Insufficient Standardization**:
  - Flow steps.
    I want clear flow steps
  - Few, no, or difficult to understand procedures.
    I want clear API for procedures.
- More powerful procedures.

Outline

Key Features - Summary
1

- **Highly Hierarchical**

Key Features - Summary
1

- **Highly Hierarchical**
  - Build on top of other custom developed TCL packages GitHub/*/octopus

Key Features - Summary
1

- **Highly Hierarchical**
  - Build on top of other custom developed TCL packages GitHub/*/octopus
  - Wraps and Extends standard RC commands (overloading allowed?).
    Introduction of new procedures.

Key Features - Summary
1

- **Highly Hierarchical**
  - Build on top of other custom developed TCL packages GitHub/*/octopus
  - Wraps and Extends standard RC commands (overloading allowed?).
    Introduction of new procedures.
- Design Maturity Level. Influences flow/attributes/reporting/etc.

Key Features - Summary
1

- **Highly Hierarchical**
  - Build on top of other custom developed TCL packages GitHub/*/octopus
  - Wraps and Extends standard RC commands (overloading allowed?).
    Introduction of new procedures.
- Design Maturity Level. Influences flow/attributes/reporting/etc.
- Follows the recommended Cadence RC synthesis flow.

Key Features - Summary
1

- **Highly Hierarchical**
  - Build on top of other custom developed TCL packages GitHub/*/octopus
  - Wraps and Extends standard RC commands (overloading allowed?).
    Introduction of new procedures.
- Design Maturity Level. Influences flow/attributes/reporting/etc.
- Follows the recommended Cadence RC synthesis flow.
- Split of common from configurable TCL

Key Features - Summary
1

- **Highly Hierarchical**
  - Build on top of other custom developed TCL packages GitHub/*/octopus
  - Wraps and Extends standard RC commands (overloading allowed?).
    Introduction of new procedures.
- Design Maturity Level. Influences flow/attributes/reporting/etc.
- Follows the recommended Cadence RC synthesis flow.
- Split of common from configurable TCL
  - Goal is to minimize the configurable part without jeopardising the flexibility

Key Features - Summary
1

- **Highly Hierarchical**
  - Build on top of other custom developed TCL packages GitHub/*/octopus
  - Wraps and Extends standard RC commands (overloading allowed?).
    Introduction of new procedures.
- Design Maturity Level. Influences flow/attributes/reporting/etc.
- Follows the recommended Cadence RC synthesis flow.
- Split of common from configurable TCL
  - Goal is to minimize the configurable part without jeopardising the flexibility
  - Design configuration file contains both TCL variables and TCL code

Key Features Summary
2

- **Automatic SDC Constraints Generation**.
  DfT setup automatically translated from NXP specific information to SDC constraints.

Key Features Summary
2

- **Automatic SDC Constraints Generation**.
  DfT setup automatically translated from NXP specific information to SDC constraints.
- **Automatic DfT setup from SDC constraints**.
  Crawling RC database for set_case_analysis and clocks.

Key Features Summary
2

- **Automatic SDC Constraints Generation**.
  DfT setup automatically translated from NXP specific information to SDC constraints.
- **Automatic DfT setup from SDC constraints**.
  Crawling RC database for set_case_analysis and clocks.
- CPF based flow.

Key Features Summary
2

- **Automatic SDC Constraints Generation**.
  DfT setup automatically translated from NXP specific information to SDC constraints.
- **Automatic DfT setup from SDC constraints**.
  Crawling RC database for set_case_analysis and clocks.
- CPF based flow.
  - libraries read via CPF

Key Features Summary
2

- **Automatic SDC Constraints Generation**.
  DfT setup automatically translated from NXP specific information to SDC constraints.
- **Automatic DfT setup from SDC constraints**.
  Crawling RC database for set_case_analysis and clocks.
- CPF based flow.
  - libraries read via CPF
  - constraints

Key Features Summary
2

- **Automatic SDC Constraints Generation**.
  DfT setup automatically translated from NXP specific information to SDC constraints.
- **Automatic DfT setup from SDC constraints**.
  Crawling RC database for set_case_analysis and clocks.
- CPF based flow.
  - libraries read via CPF
  - constraints
  - power intent

Key Features Summary
2

- **Automatic SDC Constraints Generation**.
  DfT setup automatically translated from NXP specific information to SDC constraints.
- **Automatic DfT setup from SDC constraints**.
  Crawling RC database for set_case_analysis and clocks.
- CPF based flow.
  - libraries read via CPF
  - constraints
  - power intent
- Custom colourful messages tracing for alerting future flow users.

Key Features Summary
2

- **Automatic SDC Constraints Generation**.
  DfT setup automatically translated from NXP specific information to SDC constraints.
- **Automatic DfT setup from SDC constraints**.
  Crawling RC database for set_case_analysis and clocks.
- CPF based flow.
  - libraries read via CPF
  - constraints
  - power intent
- Custom colourful messages tracing for alerting future flow users.
- Command line startup with user customizable options

Key Features Summary
2

- **Automatic SDC Constraints Generation**.
  DfT setup automatically translated from NXP specific information to SDC constraints.
- **Automatic DfT setup from SDC constraints**.
  Crawling RC database for set_case_analysis and clocks.
- CPF based flow.
  - libraries read via CPF
  - constraints
  - power intent
- Custom colourful messages tracing for alerting future flow users.
- Command line startup with user customizable options
- Source available at GitHub/*/rtlcompiler

Outline

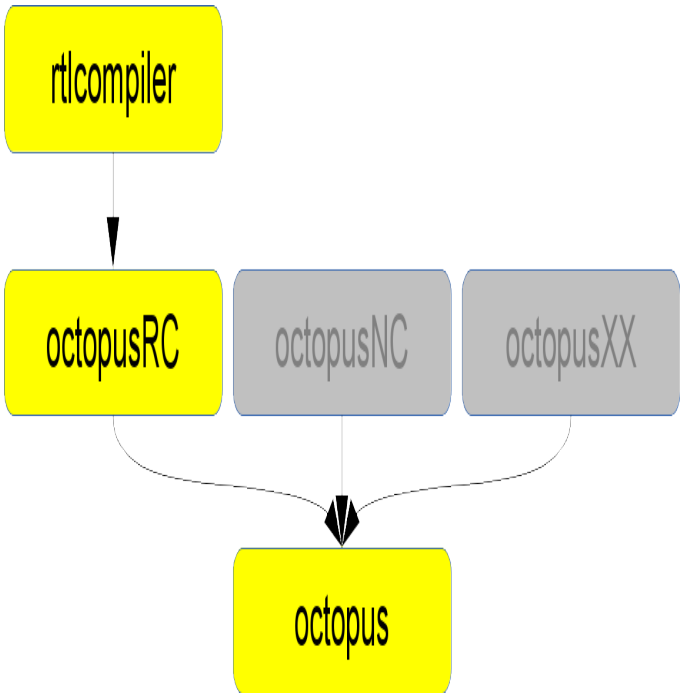
Highly Hierarchical
main.tcl (1)

```
include ./design_specific_input.tcl

# Setting RC attributes.
# Design maturity dependent
include rc_attributes.tcl

# Creating directories, cleaning files etc.
include house_keeping.tcl

#Library (including dont_use) and ple setup"
read_cpf -library $_CPF_FILE

include dont_use.tcl

include ple_setup.tcl

# Read, Elaborate and Check the Design"
include read_hdl.tcl

::octopusRC::elaborate

# Generate automatic constraints
include generate_constraints.tcl

#Read CPF in:
#power information, modes and constraints
::octopusRC::read_cpf --cpf $_CPF_FILE

#Define DFT and clock gating"
include dft_settings.tcl

include clock_gating_settings.tcl

#Synthesizing to generic
include design_constraints.tcl

# synthesize. Different levels are picked
# based on design maturity level
::octopusRC::synthesize -to_generic

#Synthesizing to gates
::octopusRC::synthesize -to_mapped

#Connect scan chains
include connect_scan_chains.tcl

#Incremental Synthesis
include design_constraints_incremental.tcl

::octopusRC::delete_unloaded_undriven

::octopusRC::synthesize -to_mapped -incremental
# Commit cpf
include ./commit_cpf.tcl
```

Highly Hierarchical
main.tcl (2)

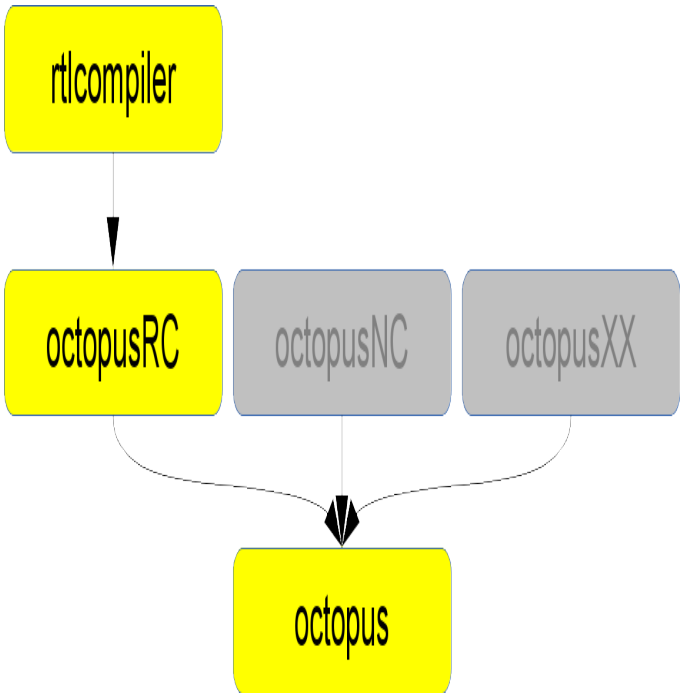
```
#!/bin/sh
# the next line restarts using -*-Tcl-*-sh \
exec rc -64 -logfile rc.log -cmdfile rc.cmd -overwrite -f "$0" -execute "set argv \"\" ; set argv ${1+\"$

#This is the main RC script. It will source other files

if { [info exists env(OCTOPUS_INSTALL_PATH) ] } {
        lappend auto_path $env(OCTOPUS_INSTALL_PATH)
} else {
        puts "ERROR: Please set environmental variable OCTOPUS_INSTALL_PATH to point to the location of oc
        exit 1
}


package require octopusRC 0.1
package require octopus   0.1

#::octopus::set_octopus_color --disable

set EXEC_PATH "" ; set DATA_PATH "" ; set CRT_LIB "" ; set CRT_CELL "" ; regexp {(.*/data/)([^/]+_lib)/([
::octopus::add_option --name "--maturity-level" --valid-values "pre-alpha alpha beta release-candidate fi
::octopus::add_option --name "--design" --variable-name "DESIGN" --default "$CRT_CELL" --help-text "Top le
::octopus::add_option --name "--reports-path" --variable-name "_REPORTS_PATH" --default "[exec pwd]/rpt" -
::octopus::add_option --name "--netlist-path" --variable-name "_NETLIST_PATH" --default "${DATA_PATH}/${CR
::octopus::add_option --name "--cpf" --variable-name "_CPF_FILE" --default "${DATA_PATH}/${CRT_LIB}/${CRT_
::octopus::add_option --name "--clean-rpt" --type "boolean" --default "false" --help-text "Clean all repor
::octopus::add_option --name "--run-speed" --default "slow" --valid-values "slow fast" --help-text "If set

set ::octopus::prog_name $prog_name

::octopus::extract_check_options_data

::octopus::abort_on error --display-help
```

TCL packages
In a nutshell



- **rtlcompiler**
- **octopusRC**
- **octopus**

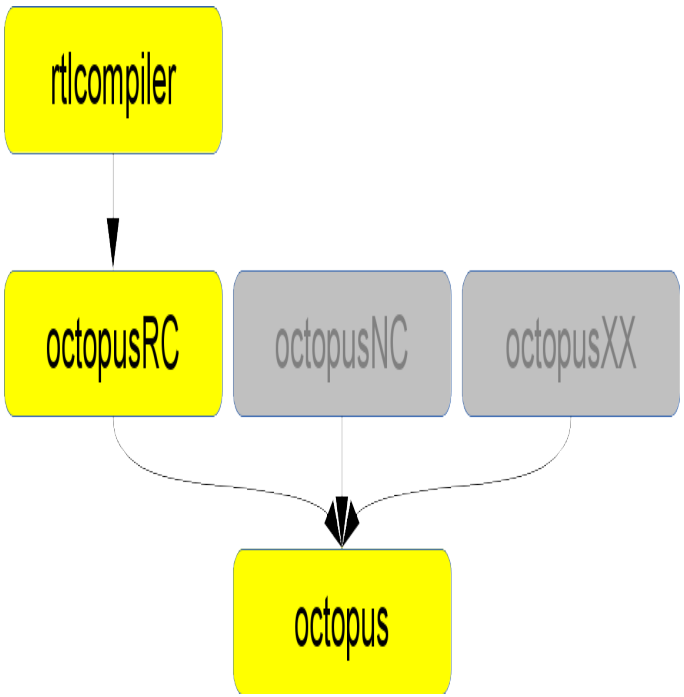
TCL packages
In a nutshell



- **rtlcompiler**

  Synthesis scripts (e.g. main.tcl).
- **octopusRC**
- **octopus**

TCL packages
In a nutshell

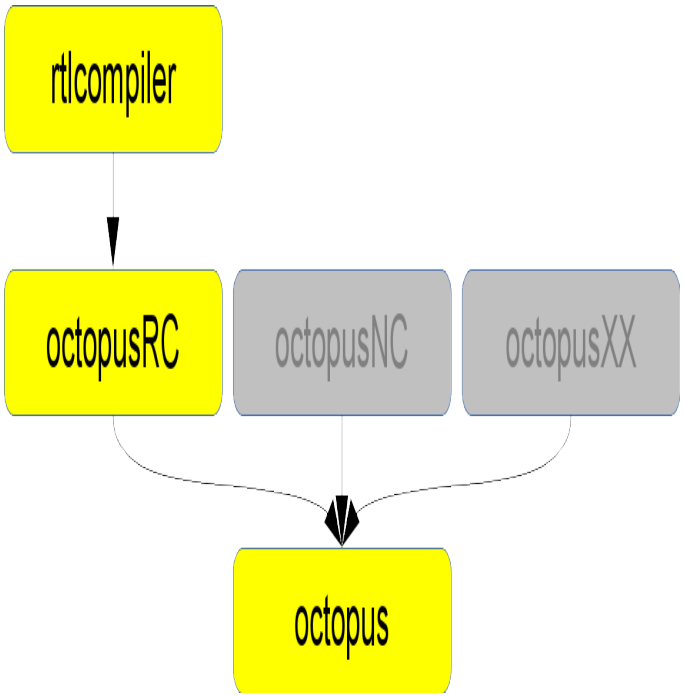


- **rtlcompiler**

  Synthesis scripts (e.g. main.tcl).
- **octopusRC**

  RTL Compiler TCL package.
  Depends on octopus package
- **octopus**

TCL packages
In a nutshell



- **rtlcompiler**

  Synthesis scripts (e.g. main.tcl).
- **octopusRC**

  RTL Compiler TCL package.
  Depends on octopus package
- **octopus**

  TCL package: procedures to deals with argument parsing, error handling, colours, consistent look and feel throughout the flow, automatic help generation.

OctopusRC package
Few Procedures (1)

- **::octopusRC::delete_unloaded_undriven**
- **::octopusRC::read_dft_abstract_model**
- **::octopusRC::define_dft_test_clocks**
- **::octopusRC::define_dft_test_signals**

OctopusRC package
Few Procedures (1)

- **::octopusRC::delete_unloaded_undriven**

  Deletes the unloaded and undriven. Depending on the design maturity this command is activated or not
- **::octopusRC::read_dft_abstract_model**
- **::octopusRC::define_dft_test_clocks**
- **::octopusRC::define_dft_test_signals**

OctopusRC package
Few Procedures (1)

- **::octopusRC::delete_unloaded_undriven**

  Deletes the unloaded and undriven. Depending on the design maturity this command is activated or not
- **::octopusRC::read_dft_abstract_model**

  Reads CTL files and applies the information to all instances, avoiding name conflicts
- **::octopusRC::define_dft_test_clocks**
- **::octopusRC::define_dft_test_signals**

OctopusRC package
Few Procedures (1)

- **::octopusRC::delete_unloaded_undriven**

  Deletes the unloaded and undriven. Depending on the design maturity this command is activated or not
- **::octopusRC::read_dft_abstract_model**

  Reads CTL files and applies the information to all instances, avoiding name conflicts
- **::octopusRC::define_dft_test_clocks**

  Define DfT clocks from the clocks specified in a certain timing mode.
- **::octopusRC::define_dft_test_signals**

OctopusRC package
Few Procedures (1)

- **::octopusRC::delete_unloaded_undriven**

  Deletes the unloaded and undriven. Depending on the design maturity this command is activated or not
- **::octopusRC::read_dft_abstract_model**

  Reads CTL files and applies the information to all instances, avoiding name conflicts
- **::octopusRC::define_dft_test_clocks**

  Define DfT clocks from the clocks specified in a certain timing mode.
- **::octopusRC::define_dft_test_signals**

  Define DfT signals from set_case_analysis.

OctopusRC package
Few Procedures(2)

- **::octopusRC::synthesize**
- **::octopusRC::set_attribute_recursive**

[1] Not fully implemented

OctopusRC package
Few Procedures(2)

- **::octopusRC::synthesize**

  Additionally writes lec, db, reports, etc.
- **::octopusRC::set_attribute_recursive**

[1] Not fully implemented

OctopusRC package
Few Procedures(2)

- **::octopusRC::synthesize**

  Additionally writes lec, db, reports, etc.
- **::octopusRC::set_attribute_recursive**

  Ideally should set attributes recursively to any type of object.[1]

[1] Not fully implemented

OctopusRC package
Few Procedures(2)

- **::octopusRC::synthesize**

  Additionally writes lec, db, reports, etc.
- **::octopusRC::set_attribute_recursive**

  Ideally should set attributes recursively to any type of object.[1]
- others ...

[1] Not fully implemented

Design Maturity Level
1

- **::octopusRC::set_design_maturity_level**
  pre-alpha, alpha, beta, release-candidate, final.

Design Maturity Level
1

- **::octopusRC::set_design_maturity_level**
  pre-alpha, alpha, beta, release-candidate, final.
- RC attributes are differentiated as shown on slide 16.

Design Maturity Level
1

- **::octopusRC::set_design_maturity_level**
  pre-alpha, alpha, beta, release-candidate, final.
- RC attributes are differentiated as shown on slide 16.
- Synthesis optimization efforts are set at medium or high.

Design Maturity Level
1

- **::octopusRC::set_design_maturity_level**
  pre-alpha, alpha, beta, release-candidate, final.
- RC attributes are differentiated as shown on slide 16.
- Synthesis optimization efforts are set at medium or high.
- Boundary optimization settings.

Design Maturity Level
2

```
#TABLE HEADERS DESCRIPTION
#rc_attribute                          pre-alpha       alpha           beta    release-candidate       final
show_report_options                    true            true            true            true            true
detailed_sdc_messages                  true            true            true            true            true
group_generate_portname_from_netname   true            true            true            true            true
lp_insert_clock_gating                 -               true            true            true            true
lp_insert_clock_gating_incremental     -               true            true            true            true
write_vlog_bit_blast_constants         true            true            true            true            true
information_level                      9               9               9               9               9
continue_on_error                      true            -               -               -               -
find_inefficient_use                   true            true            true            -               -
log_command_error                      true            true            true            -               -
report_tcl_command_error               true            true            true            -               -
source_verbose                         true            true            -               -               -
source_verbose_proc                    true            true            -               -               -
source_verbose_info                    -               false           -               -               -
source_suspend_on_error                -               true            true            -               -
hdl_report_case_info                   true            true            true            -               -
hdl_track_filename_row_col             true            true            true            -               -
hdl_array_naming_style                 %s_%d_          %s_%d_          %s_%d_          %s_%d_          %s_%d_
hdl_instance_array_naming_style        %s_%d_          %s_%d_          %s_%d_          %s_%d_          %s_%d_
hdl_record_naming_style                %s_%s_          %s_%s_          %s_%s_          %s_%s_          %s_%s_
hdl_generate_index_style               %s_%d_          %s_%d_          %s_%d_          %s_%d_          %s_%d_
hdl_generate_separator                 __              __              __              __              __
#write_vlog_empty_module_for_black_box true            true            -               -               -
delete_unloaded_insts                  false           false           -               -               -
delete_unloaded_seqs                   false           false           -               -               -
...
```

Design Maturity Level
2

```
#TABLE HEADERS DESCRIPTION
#rc_attribute                          pre-alpha       alpha           beta    release-candidate       final
...
hdl_preserve_unused_registers          true            true            -               -               -
hdl_track_module_elab_memory_and_runtime true          -               true            true            -
dp_postmap_downsize                    -               -               true            true            true
dp_postmap_upsize                      -               -               true            true            true
fail_on_error_mesg                     -               -               -               true            true
boundary_optimize_invert_hier_pins     -               -               -               true            true
boundary_optimize_constant_hier_pins   false           -               -               -               -
dp_rewriting                           -               -               -               advanced        advanc
hdl_error_on_blackbox                  -               -               -               true            true
hdl_error_on_logic_abstract            -               -               -               true            true
hdl_unconnected_input_port_value       -               -               0               0               0
hdl_undriven_output_port_value         -               -               0               0               0
hdl_undriven_signal_value              -               -               0               0               0
iopt_enable_floating_output_check      true            true            true            true            true
iopt_ultra_optimization                -               -               -               true            true
glo_redrem_ultra_effort                -               -               true            true            true
optimize_constant_0_flops              false           false           -               -               -
optimize_constant_1_flops              false           false           -               -               -
optimize_constant_latches              false           false           -               -               -
optimize_merge_flops                   false           false           -               -               -
optimize_merge_latches                 false           false           -               -               -
propagate_constant_from_timing_model   false           false           -               -               -
tns_opto                               -               -               true            true            true
ultra_global_mapping                   -               -               true            true            true
dft_identify_internal_test_clocks      no_cgic_hier    no_cgic_hier    -               -               -
dft_report_empty_test_clocks           true            true            true            true            true
dft_shift_register_identification_mode logical_only    logical_only    logical_only    logical_only    logica
```

Split Common from Configurable TCL Code
1

- Group the configurable part in one file

Split Common from Configurable TCL Code
1

- Group the configurable part in one file
- Allow both TCL variables and TCL code
  Reading HDL files, LEF's, reading CTL's, scan-enable specification, scan-chains setup, constraints generation, test-point insertion

Configuration File
1

```
namespace eval diehardus {
# File containing the RTL files. Supported formats rc,text,utel
variable read_hdl                ":::octopusRC::read_hdl --type rc ../rtlcompiler/cmd/read_hdl.tcl"

variable lefs "
                $env(CADENV_HOME)/.caddata/krfdi/tools/cadence_edi/xkrfdix_5.6.lef\
...

variable read_ctl {
                ::octopusRC::read_dft_abstract_model\
                        -assume_connected_shift_enable\
                        --ctl \
                                $env(PROJECT_WORK)/data/hrxc_hrxc1_lib/hrxc_hrxc1/catviews/hrxc_hrxc1_ana_tp
                                $env(PROJECT_WORK)/data/hrxc_hrxc2_lib/hrxc_hrxc2/catviews/hrxc_hrxc2_ana_tp
                                $env(PROJECT_WORK)/data/hrxc_hrxc2_lib/hrxc_hrxc2/catviews/hrxc_hrxc2_trim_t
                                ... \
                        --boundary-opto \
                        --debug-level 2

                ::octopusRC::read_dft_abstract_model \
                        --ctl \
                                $env(PROJECT_WORK)/data/ltthf_lib/ltthf_asdopefsd_asdcsda/TEST/ltthf_asdopef
                        --debug-level 2
                }
...
}
```

Configuration File
2

```
...
set scan_chains_insertion {
                # Define floating segment with falling edge FF's. They will be put in front of the chain.
                define_dft floating_segment \
                        -name falling_edge_flops [::octopus::find_fall_edge_objects]

                ## Define scan chains
                define_dft scan_chain \
                        -name allFF_0 \
                        -sdi I0/u0_hrxc_ic_core_test/si[0] \
                        -sdo I0/u0_hrxc_ic_core_test/so[0] \
                        -non_shared_output \
                        -head falling_edge_flops \
                        -terminal_lockup level_sensitive

                define_dft scan_chain \
                        -name allFF_1 \
                        -sdi I0/u0_hrxc_ic_core_test/si[1] \
                        -sdo I0/u0_hrxc_ic_core_test/so[1] \
                        -non_shared_output \
                        -terminal_lockup level_sensitive

                connect_scan_chains -incremental

                fix_scan_path_inversions allFF_0
                fix_scan_path_inversions allFF_1
}
...
```

Automatic SDC Constraints Generation

- NXP DfT: Use of serial shift registers, containing test-data file describing the static values during test.

Automatic SDC Constraints Generation

- NXP DfT: Use of serial shift registers, containing test-data file describing the static values during test.
- By parsing these test data files, already a big chunk of constraints can be automatically generated.

Automatic SDC Constraints Generation

- NXP DfT: Use of serial shift registers, containing test-data file describing the static values during test.
- By parsing these test data files, already a big chunk of constraints can be automatically generated.
- Generated constraints used at least in three timing modes **scan-shift**, **capture** and **functional**.

Automatic DfT Setup from SDC Constraints

- **Clocks** ::octopusRC::define_dft_test_clocks
- **Test Signals** ::octopusRC::define_dft_test_signals

Automatic DfT Setup from SDC Constraints

- **Clocks** ::octopusRC::define_dft_test_clocks
  - Specify the timing mode
- **Test Signals** ::octopusRC::define_dft_test_signals
  - Specify the timing mode

Automatic DfT Setup from SDC Constraints

- **Clocks** ::octopusRC::define_dft_test_clocks
  - Specify the timing mode
  - Clocks are extracted from RC database
- **Test Signals** ::octopusRC::define_dft_test_signals
  - Specify the timing mode
  - Extract the set_case_analysis from RC database

Automatic DfT Setup from SDC Constraints

- **Clocks** ::octopusRC::define_dft_test_clocks
  - Specify the timing mode
  - Clocks are extracted from RC database
  - Clocks can be added or removed (not recommended).
- **Test Signals** ::octopusRC::define_dft_test_signals
  - Specify the timing mode
  - Extract the set_case_analysis from RC database
  - Signals to add or remove (not recommended)

Outline

Conclusion
Things to Remember

- RTL Compiler Synthesis flow

Conclusion
Things to Remember

- RTL Compiler Synthesis flow
- Highly hierarchical (scripts and two other TCL packages)

Conclusion
Things to Remember

- RTL Compiler Synthesis flow
- Highly hierarchical (scripts and two other TCL packages)
- Customization part of the flow is in a single file and contains both variables and code

Conclusion
Things to Remember

- RTL Compiler Synthesis flow
- Highly hierarchical (scripts and two other TCL packages)
- Customization part of the flow is in a single file and contains both variables and code
- Generate DfT related SDC from NXP specific test-data files

Conclusion
Things to Remember

- RTL Compiler Synthesis flow
- Highly hierarchical (scripts and two other TCL packages)
- Customization part of the flow is in a single file and contains both variables and code
- Generate DfT related SDC from NXP specific test-data files
- Generate DfT constraints from SDC.

Conclusion
Things to Remember

- RTL Compiler Synthesis flow
- Highly hierarchical (scripts and two other TCL packages)
- Customization part of the flow is in a single file and contains both variables and code
- Generate DfT related SDC from NXP specific test-data files
- Generate DfT constraints from SDC.
- CPF only flow

Conclusion
Things to Remember

- RTL Compiler Synthesis flow
- Highly hierarchical (scripts and two other TCL packages)
- Customization part of the flow is in a single file and contains both variables and code
- Generate DfT related SDC from NXP specific test-data files
- Generate DfT constraints from SDC.
- CPF only flow
- Flow and packages freely available at GitHub
  Can be extended as needed (e.g. technology exploration)

Questions?