cadence

Lab Instructions: Parallel Distributed Interactive MMMC Analysis And Tempus ECO (Paradime Flow) in Common User Interface (CUI)

Rapid Adoption Kit (RAK)

Product Version: Tempus 20.1

March 2021

Contents

Introduction

The Paradime flow in CUI allows you to perform various tasks, such as distributed timing analysis, ECO, and any interactive debugging (such as reporting and what-if analysis) in a single Tempus session leading to improved usability of the tool.

The first lab will cover the Restored Analysis flow, and the second lab will cover the Full Analysis flow.

It is recommended to run the Labs in the order described in this document.

Note: The Paradime feature in CUI requires a Tempus ECO license and a Common UI license.

Terms

ECO	Engineering Change Order
Tempus	Cadence SignOff STA tool
AAE	Name of the delay calculation engine (Advanced Analysis Engine)
MMMC	Multi-Mode Multi Corner
View	Combination of a corner and a timing mode
DB	Database
DMMMC	Distributed MMMC timing analysis environment in Tempus
DRV	Design Rule Violation, such as max_cap and max_tran.
STA	Static Timing Analysis
PBA	Path-Based Analysis

Paradime Top-Level Flow

The Paradime solution uses the DMMMC infrastructure of Tempus and sets up a distributed session environment for a large number of views to perform analysis/optimization, which works like a regular Tempus shell. It provides you with one master cockpit to manage optimization and analysis.

The following steps describe the top-level flow:

a) The Tempus Master spawns client jobs. You need to specify the design data to be loaded on clients. The same design data is used to initialize the master and clients. The analysis commands run on clients, and ECO optimization runs on the master.

   The same Tempus session is usable for both timing analysis and ECO runs. This avoids separate runs for the timing analysis, ECO, post-ECO analysis, or interactive debugging.

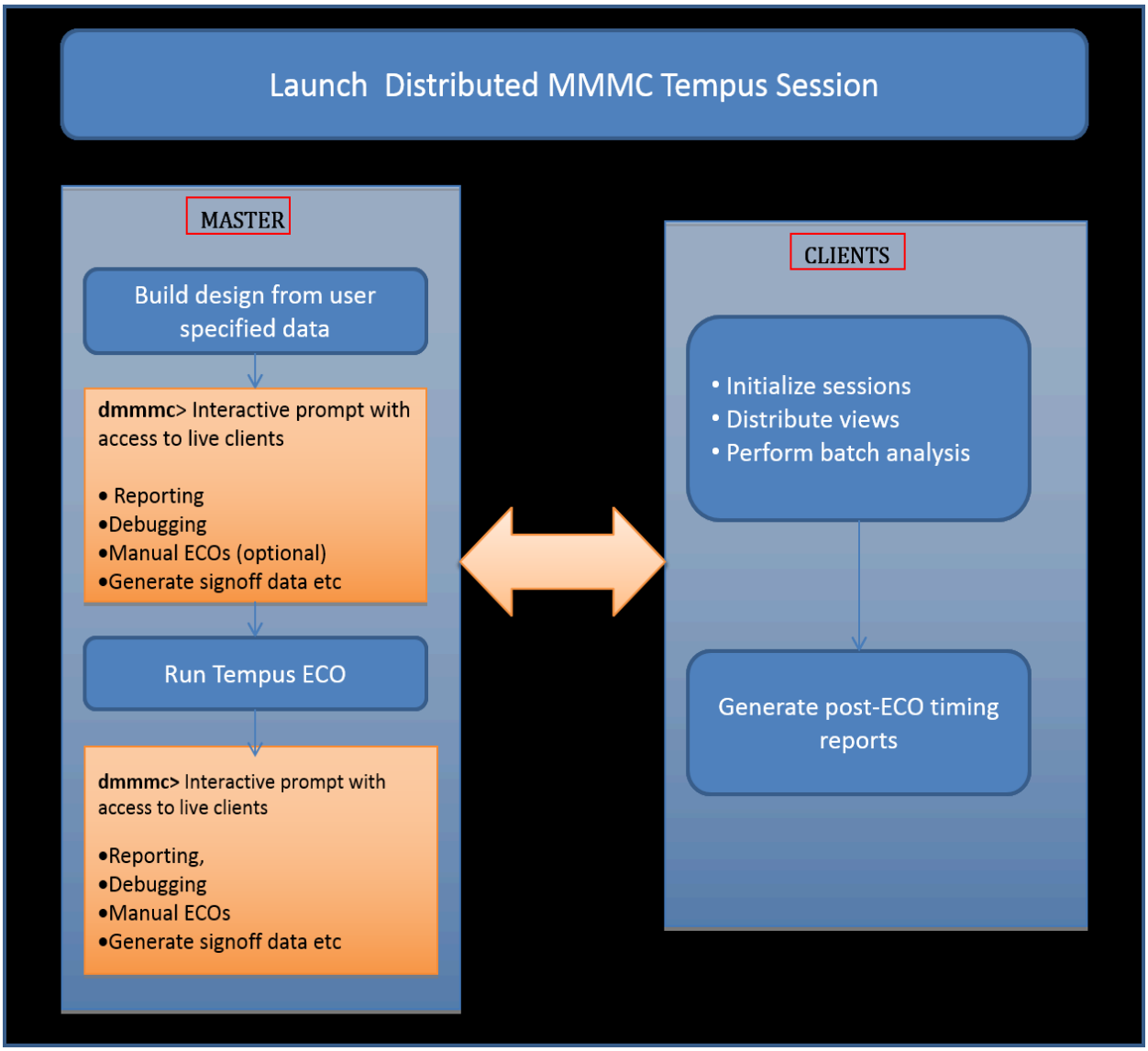
b) A DMMMC interactive prompt is displayed to run any commands before ECO.

c) Run ECO. Towards the end of the ECO run, the software will automatically source the ECO file on all clients and generate the post-ECO timing data. This will avoid any miscorrelation issues, as seen in the post-ECO timing summary in the Tempus ECO run and the timing seen in the Tempus signoff run after sourcing ECOs generated by ECO.

d) Post the ECO run, the Tempus master shell is available and accessible to Tempus clients. This can be used for interactive debugging/what-if analysis/reporting, like the regular Tempus shell.

e) In the same session, you can perform multiple ECO runs if needed.

The following flow diagram describes the flow pictorially:



How to Access the RAK Database

You can find the RAK testcase database, Scripts, and References at the ‘Attachments’ and ‘Related Solutions’ sections below the PDF.

You can search for this PDF with the document 'Title' on https://support.cadence.com.

Directory Structure

The lab directory structure is as follows:

```
< Paradime_RAK/>
            design/
            docs/
            libs/
            scripts/
            work/
            common_UI/
```

Tools and Scripts

This RAK uses Tempus Digital Implementation. Make sure you have the **Tempus 20.1** version installed.

Note: You must use the same tool version and environment (Tcl, libraries, constraints, and so on) throughout the labs.

Typographical Conventions

Throughout this RAK, the required user input is indicated by a typical command prompt. For instance:

```
➢ cd Paradime_RAK/common_UI/work/
```

Design

The sample test case is a generic DTMF design containing the following:

- Approximately 10k instances
- Sequential, combinational, and state machine logic
- Hard macros: PLL, RAM, and ROM
- Seven clocks
- 36 Views (2 libraries x 6 corners x 3 modes):

- Libraries are Worst and Typical
- Corners are Cmax, RCmax, Rmax, Cmin, RCmin, and Rmin
- Modes are Functional, Test, and Scan

Note: This design is mapped on Cadence open source libraries. This explains why the clock period and cell delay are probably larger than usual.

License/Resource Requirement of the Flow

The licensing requirements for this flow require Tempus to be invoked with the `-eco` and `-common_ui` license options for performing optimization in addition to the Tempus Distributed Multi-Mode Multi-Corner licensing requirements.

Flow Description

The flow requires one top-level script to be provided to the Tempus master. This script contains settings for distributed/multi-CPU configuration, design loading commands, ECO settings/commands and reporting commands, and so on.

To enable this flow, you need to make the following settings in the script provided to the Tempus Master:

```
set_distributed_mmmc_mode -eco true
```

In this flow, the number of remote hosts should be equal to the number of views.

The setup for Tempus master and clients is done upfront. Both the master and clients load the design data for either a full analysis or the restored analysis flow.

There are several ways to provide input data in a Paradime flow:

a) Full analysis flow by loading data from scratch: You can use this mode to run Paradime on any input netlist by providing a data pointer using the Tempus atomic commands.

b) Restored analysis flow using the Tempus saved session: In this mode, Tempus clients will not be required to perform full-timing updates since timing data will come from the saved session for each active view.

The Tempus master is used for an ECO run, and clients are used for timing analysis runs. All data configuration settings should be done using the MMMC infrastructure.

Required Input Files

Design Data	
LEF	Captures physical library rules for place and route and the abstract of the cells, such as the technology layer, design rules, and metal cap.
Timing Libraries	Captures characteristics of cells, including function, slew, delay, timing, power, and noise at the given operating conditions.
Netlist	Gate-level netlist provides the logical connectivity of the design.
SDC	This is a common format (Synopsis Design Constraints format) for constraining the design, which is essential to meet the design requirements in terms of Area, Timing, and Power.
DEF	Captures all the physical aspects of a design, including the floor planning information, such as the die size, connectivity, and physical location of the cell.
SPEF	SPEF is an IEEE format that captures the RC values required for the actual net and cell delay calculations.

Different Combinations of Flows

There are several possible combinations for master and clients. You can run the flow in the following modes:

Flavors	**Client**	**Master**	**Support**
1	Logical	Physical	Supported in the Restored flow only
2	Logical	Logical	Supported in both (Full Analysis and Restored flow)
3	Physical	Physical	Supported in both (Full Analysis and Restored flow)

LAB 1: Restored Analysis Flow

Overview: The saved sessions from individual STA runs are restored in Paradime. These individual SMSC STA runs have unique view names and are saved in the MMMC configuration using a view definition file with a unique library set, delay corner, RC corner, constraint mode, analysis view, and so on. This flow uses the saved SMSC sessions.

Goal: To perform merging of reports after the full timing analysis run or run manual commands to find out critical views of interest and perform interactive querying and debugging on one, few, or all client views.

Lab Flow

The flow consists of the following steps:

- Restored Analysis Flow Using Physical Master and Logical Client
- Tempus ECO Optimization
- Restore Analysis Flow Using Logical Master and Logical Client
- Tempus ECO Optimization
- Restore Analysis Flow Using Physical Master and Physical Client
- Tempus ECO Optimization

LAB 1.1: Restored Analysis Flow Using Physical Master and Logical Client

Step 1. Design loading in Paradime

The design data is restored in Paradime from the logical STA saved session. In this mode, Tempus clients are not required to perform full timing updates because the timing data will come from the saved session for each active view. The same number of clients is required as the number of sessions to be restored. Each saved session can have the data for one MMMC view. Data from the same restored sessions is loaded on master for all views together. The physical data of the design is read on the master.

A snippet of Tcl scripts launched for the timing and Paradime session is given below.

- `cd Paradime_RAK/common_UI/work`

How to run:

```
tempus -nowin -init ../scripts/sta_view1_logical.tcl.cui -
common_ui
```

- `cat ../scripts/sta_view1_logical.tcl.cui`

```
set_multi_cpu_usage -local_cpu 16
read_mmmc ../work/STYLUS_DB/viewDefinition_view1.tcl
read_netlist
"../../design/ECO_INIT_11_optSetup.enc.dat/dtmf_recvr_core.v.gz" -
top dtmf_recvr_core
init_design
read_spef -rc_corner corner_worst_RCMAX
"../../design/SPEF/corner_worst_RCMAX.spef.gz"
write_db view1_logical_cui
```

How to run:

```
tempus -nowin -init ../scripts/sta_view2_logical.tcl.cui -
common_ui
```

- `cat ../scripts/sta_view2_logical.tcl.cui`

```
set_multi_cpu_usage -local_cpu 16
read_mmmc ../work/STYLUS_DB/viewDefinition_view2.tcl
read_netlist
"../../design/ECO_INIT_11_optSetup.enc.dat/dtmf_recvr_core.v.gz" -top
dtmf_recvr_core
init_design
read_spef -rc_corner corner_worst_RCMIN
"../../design/SPEF/corner_worst_RCMIN.spef.gz"
write_db view2_logical_cui
```

How to run:

```
tempus -nowin -init ../scripts/paradime_restoreFlow.tcl.cui -
eco -common_ui
```

- `cat ../scripts/paradime_restoreFlow.tcl.cui`

```
set_distributed_hosts -local -time_out 60
set_multi_cpu_usage -local_cpu 4 -remote_host 2 -cpu_per_remote_host
2
set_distributed_mmmc_mode -eco_merge_mmmc -eco true
set_db eco_setup_view_list {func_slow_RCMAX func_slow_RCMIN}
set_db eco_hold_view_list {func_slow_RCMAX func_slow_RCMIN}
set_db eco_def_file_list "../../design/full.def"
set_db eco_lef_file_list
"../../libs/lef/FreePDK45_lib_v1.0.lef ../../libs/MACRO/LEF/pllclk.le
f
../../libs/MACRO/LEF/ram_256X16A.lef ../../libs/MACRO/LEF/rom_512x16
A.lef"
distribute_read_db -read_db {view1_logical_cui view2_logical_cui} -
out_dir restore_mode1_cui
set_db opt_signoff_retime path_slew_propagation
set_db opt_signoff_allow_multiple_incremental true
set_db opt_signoff_verbose true
eval_legacy {set_eco_opt_mode -write_eco_for_logical_client true}
opt_signoff -setup
opt_signoff -hold
opt_signoff -drv
opt_signoff -power
opt_signoff -area
```

1. Specify the multiple-CPU processing configuration for the distributed processing.

The `set_multi_cpu_usage` command specifies the number of threads to use for multithreading or the maximum number of computers to use for the distributed processing.

Tcl command:

```
set_distributed_hosts -local -time_out 60
set_multi_cpu_usage -local_cpu 4 -remote_host 2 -
cpu_per_remote_host 2
```

2. Specify the command given below with the `-eco` parameter to enable the flow. Use the `-eco_merge_mmmc` parameter to create the ECO analysis view(s) from the view definition files in restore sessions.

Tcl command:

```
set_distributed_mmmc_mode -eco_merge_mmmc -eco true
```

3. Specify a list of setup and hold views for Tempus ECO (if different than the views restored on clients). By default, all the clients' views will be in the setup/hold view list.

Tcl command:

```
set_db eco_setup_view_list {func_slow_RCMAX func_slow_RCMIN}
set_db eco_hold_view_list {func_slow_RCMAX func_slow_RCMIN}
```

4. Specify the following command to launch multiple Tempus DMMMC clients to restore the design data:

Tcl command:

```
distribute_read_db -read_db {view1_logical_cui view2_logical_cui}
-out_dir restore_mode1_cui
```

5. Specify the list of DEF/LEF files required for the physical Tempus ECO run.

Tcl command:

```
set_db eco_def_file_list "../../design/full.def"
set_db eco_lef_file_list
"../../libs/lef/FreePDK45_lib_v1.0.lef ../../libs/MACRO/LEF/pllclk
.lef
../../libs/MACRO/LEF/ram_256X16A.lef ../../libs/MACRO/LEF/rom_512x
16A.lef"
```

Step 2: Tempus ECO Optimization

After the design loading and analysis (relevant only in the full analysis flow) is done on clients, the interactive prompt is displayed.

Any prerequisite commands/setting for the ECO run should be applied by this stage. The ECO file written by optimization is sent automatically to the clients for the post-ECO signoff timing report generation.

ECO DB generation and its usage for Tempus ECO will be transparent to you in this flow.

After the data restoration on clients in the `restoreDesign` flow, the software will start an automatic generation of ECO DB for all views. By default, it will generate GBA ECO DBs.

On completion of the Tempus ECO run, the master will return to the interactive prompt with access to Tempus clients for signoff timing report generation and manual what-if analysis. You can also do successive Tempus ECO runs.

The following settings are applied for optimization:

```
set_db opt_signoff_retime path_slew_propagation
set_db opt_signoff_allow_multiple_incremental true
set_db opt_signoff_verbose true
eval_legacy {set_eco_opt_mode -write_eco_for_logical_client true}
```

1. To support PBA-based optimization for all optimizers

Tcl command:

```
set_db opt_signoff_retime path_slew_propagation
```

2. To allow multiple types of optimization in an incremental mode

In the Paradime flow, in a non-incremental optimization mode, the software generates ECO DB twice for each `opt_signoff` run; the first ECO DB before running the optimization and the second ECO DB after optimization to generate post ECO timing. If you want to run multiple ECO runs, it is recommended to run the flow in an incremental optimization mode to reduce the run time.

In an incremental optimization, ECO DB is generated only once before the optimization begins and the same ECO DB is updated in the memory with netlist changes for each `opt_signoff` run. Since there is no multiple ECO DB generation, it results in run time saving without any significant impact on the accuracy.

In some cases, fresh timing information is required for the next ECO run, but the user may not wish to exit the incremental mode due to the run time. This is usually required when netlist changes are significantly higher, resulting in a large number of ECOs, for example, after the power optimization. It is recommended to start the next ECO run with fresh timing information for accuracy reasons. To achieve this, a new option `-update_timing` is provided to the `opt_signoff` command.

Tcl command:

```
set_db opt_signoff_allow_multiple_incremental true
```

3. To specify whether the command should output the log file reporting with verbosity or not.

Tcl command:

```
set_db opt_signoff_verbose true
```

Fixing timing violations:

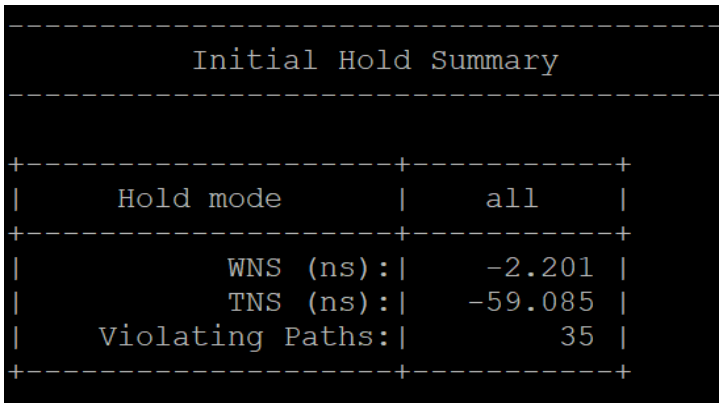
1. Hold Violations: Hold timing violations must be fixed without impacting the setup timing and without creating any design rule violations.

Specify the following command to fix hold violations:

Tcl command:

```
opt_signoff -hold
```

You can check the hold violations in the **Initial Hold Summary** section of the log file.



You can check the total number of paths fixed by Tempus ECO in the **Final Hold Summary** section of the log file (the **Violating Paths** count has reduced to **0** now):

```
----------------------------------------
         Final Hold Summary
----------------------------------------

+--------------------+-----------+
|     Hold mode      |    all    |
+--------------------+-----------+
|           WNS (ns):|     0.004 |
|           TNS (ns):|     0.000 |
|    Violating Paths:|         0 |
+--------------------+-----------+
```

From this analysis, try to answer the following questions:

- Are there hold timing violations initially?
- Have all violations been fixed?

2. Setup Violations: Setup timing violations must be fixed without impacting the hold timing and without creating any design rule violations.

Specify the following command to fix setup violations:

Tcl command:

```
opt_signoff -setup
```

```
----------------------------------------
        Initial Setup Summary
----------------------------------------

+--------------------+-----------+
|     Setup mode     |    all    |
+--------------------+-----------+
|           WNS (ns):|     1.030 |
|           TNS (ns):|     0.000 |
|    Violating Paths:|         0 |
+--------------------+-----------+
```

Setup fixing is not required as there are no setup violations.

3. DRV violations: DRV timing violations must be fixed without impacting the hold timing and setup timing violations.

Specify the following command to fix DRV violations:

Tcl command:

```
opt_signoff -drv
```

Initial DRV Summary:

DRVs	Signal nets		Clock nets	
	Nr nets(terms)	Worst Vio	Nr nets(terms)	Worst Vio
max_cap	1 (1)	-0.001	0 (0)	0.000
max_tran	5 (5)	-0.112	3 (3)	-2.663

As shown in the above table, the total number of nets with DRV violations is 9. The count has been reduced to 6.

Final DRV Summary:

DRVs	Signal nets		Clock nets	
	Nr nets(terms)	Worst Vio	Nr nets(terms)	Worst Vio
max_cap	0 (0)	0.000	0 (0)	0.000
max_tran	3 (3)	-0.112	3 (3)	-2.663

From this analysis, try to answer the following questions:

- What is the initial Signal Nets max_cap violation?
- What is the initial Signal Nets max_tran violation?
- Does the software reduce the max_tran violation count?

4. Power Recovery: The power recovery must be done without impacting any timing and design rule violations.

Specify the following command for the power recovery:

Tcl command:

```
opt_signoff -power
```

Initial Power Summary:

```
Power Summary
-------------------------------------------------------------------------------------------------------
Group               Leakage(mW)      Internal(mW)       Switching(mW)        Total(mW)     (#insts)
-------------------------------------------------------------------------------------------------------
Combinational          0.00544331       0.110704          0.224082          0.340229         7927
Sequential             0.00242555      0.0478468         0.0184299         0.0687023          579
Pad                             0              0                 0                 0            0
Block                  0.00400015        2.91847         0.0104232            2.9329            4
-------------------------------------------------------------------------------------------------------
Total Power             0.011869        3.07703          0.252935          3.34183         8510
-------------------------------------------------------------------------------------------------------
```

Final Power Summary:

```
Power Summary
-------------------------------------------------------------------------------------------------------
Group               Leakage(mW)      Internal(mW)       Switching(mW)        Total(mW)     (#insts)
-------------------------------------------------------------------------------------------------------
Combinational          0.0047096       0.0970423          0.217012          0.318763         7700
Sequential             0.00242555      0.0478357         0.0200939         0.0703551          579
Pad                             0              0                 0                 0            0
Block                  0.00400015        2.91847         0.0104106           2.93289            4
-------------------------------------------------------------------------------------------------------
Total Power (final)     0.0111353        3.06335          0.247516             3.322         8283
-------------------------------------------------------------------------------------------------------
```

5. Area Recovery: The area must be reduced without impacting the setup timing, hold timing, and without creating any design rule violations. Area recovery is performed by downsizing both the combination and sequential cells, in addition to the buffer removal.

Specify the following command for the area recovery:

Tcl command:

```
opt_signoff -area
```

```
*info:
*info:    Total 30 instances resized or VT-Swapped
*info:
*summary:     30 cell type changes
*       :     28 instances changed cell type from 'BUF_X16' to 'BUF_X8'
*       :      1 instance  changed cell type from 'BUF_X32' to 'BUF_X16'
*       :      1 instance  changed cell type from 'BUF_X8' to 'BUF_X4'
*info:       total area recovered=16.2
```

LAB 1.2: Restored Analysis Flow Using Logical Master and Logical Client

Step 1. Design loading in Paradime

The design data is restored in Paradime from the logical STA saved session, as shown in LAB1.1. In this flow, Tempus clients are not required to perform full timing updates as the timing data will come from the saved session for each active view. The same number of clients is required as the number of sessions to be restored. Each saved session can have data for one MMMC view. In this flow, the master is loaded with logical data only.

A snippet of Tcl scripts launched for the timing and Paradime session is given below.

- `cd Paradime_RAK/common_UI/work`

How to run:

```
tempus -nowin -init ../scripts/sta_view1_logical.tcl.cui -
common_ui
```

- `cat ../scripts/sta_view1_logical.tcl.cui`

```
set_multi_cpu_usage -local_cpu 16
read_mmmc ../work/STYLUS_DB/viewDefinition_view1.tcl
read_netlist
"../../design/ECO_INIT_11_optSetup.enc.dat/dtmf_recvr_core.v.gz" -
top dtmf_recvr_core
init_design
read_spef -rc_corner corner_worst_RCMAX
"../../design/SPEF/corner_worst_RCMAX.spef.gz"
write_db view1_logical_cui
```

How to run:

```
tempus -nowin -init ../scripts/sta_view2_logical.tcl.cui -
common_ui
```

- `cat ../scripts/sta_view2_logical.tcl.cui`

```
set_multi_cpu_usage -local_cpu 16
read_mmmc ../work/STYLUS_DB/viewDefinition_view2.tcl
read_netlist
"../../design/ECO_INIT_11_optSetup.enc.dat/dtmf_recvr_core.v.gz" -
top dtmf_recvr_core
init_design
read_spef -rc_corner corner_worst_RCMIN
"../../design/SPEF/corner_worst_RCMIN.spef.gz"
write_db view2_logical_cui
```

How to run:

```
tempus -nowin -init ../scripts/paradime_restoreFlow_1.tcl.cui
-eco -common_ui
```

- `cat ../scripts/paradime_restoreFlow_1.tcl.cui`

```
set_distributed_hosts -local -time_out 60
set_multi_cpu_usage -local_cpu 4 -remote_host 2 -cpu_per_remote_host
2

set_distributed_mmmc_mode -eco_merge_mmmc -eco true
set_db eco_setup_view_list {func_slow_RCMAX func_slow_RCMIN}
set_db eco_hold_view_list {func_slow_RCMAX func_slow_RCMIN}

distribute_read_db -read_db {view1_logical_cui view2_logical_cui} -
out_dir restore_mode2_cui


set_db opt_signoff_retime path_slew_propagation
set_db opt_signoff_allow_multiple_incremental true
set_db opt_signoff_verbose true
opt_signoff -setup
opt_signoff -hold
opt_signoff -drv
opt_signoff -power
opt_signoff -area
```

Step 2: Tempus ECO Optimization

After the design loading and analysis (relevant only in the full analysis flow) is done on clients, the interactive prompt is displayed.

Any prerequisite commands/setting for ECO run should be applied by this stage. ECO files written by optimization are sent automatically to clients for post-ECO signoff timing report generation.

On completion of the Tempus ECO run, the master will return to the interactive prompt with access to Tempus clients for signoff timing report generation and manual what-if analysis. You can also do successive Tempus ECO runs.

Fixing timing violations:

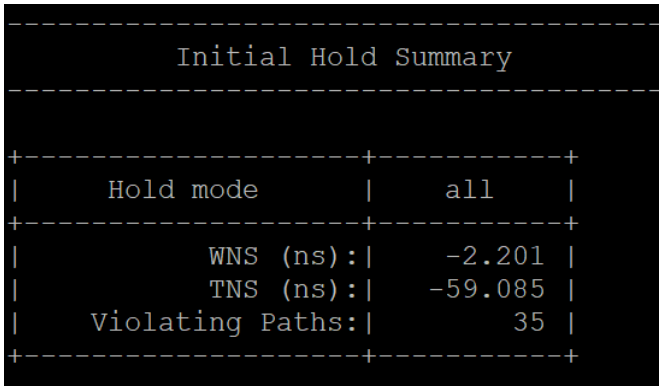
1. Hold Violations: Hold timing violations must be fixed without impacting the setup timing and without creating any design rule violations.

Specify the following command to fix hold violations:

Tcl command:

```
opt_signoff -hold
```

You can check the hold violations in the **Initial Hold Summary** section of the log file.



You can check the total number of paths fixed by Tempus ECO in the **Final Hold Summary** of the log file (the **Violating Paths** count has reduced to **0** now):

```
----------------------------------------
         Final Hold Summary
----------------------------------------

+--------------------+-----------+
|     Hold mode      |    all    |
+--------------------+-----------+
|           WNS (ns):|     0.013 |
|           TNS (ns):|     0.000 |
|    Violating Paths:|         0 |
+--------------------+-----------+
```

From this analysis, try to answer the following questions:

- Are there hold timing violations initially?
- Have all violations been fixed?

2. Setup Violations: Setup timing violations must be fixed without impacting the hold timing and without creating any design rule violations**.**

Specify the following command to fix setup violations:

Tcl command:

```
opt_signoff -setup
```

```
----------------------------------------
        Initial Setup Summary
----------------------------------------

+--------------------+-----------+
|     Setup mode     |    all    |
+--------------------+-----------+
|           WNS (ns):|     1.030 |
|           TNS (ns):|     0.000 |
|    Violating Paths:|         0 |
+--------------------+-----------+
```

Setup fixing is not required as there are no setup violations.

3. DRV Violations: DRV timing violations must be fixed without impacting the hold timing and setup timing violations.

Specify the following command to fix DRV violations:

Tcl command:

```
opt_signoff -drv
```

Initial DRV Summary:

DRVs	Signal nets		Clock nets	
	Nr nets(terms)	Worst Vio	Nr nets(terms)	Worst Vio
max_cap	1 (1)	-0.001	0 (0)	0.000
max_tran	5 (5)	-0.112	3 (3)	-2.663

As mentioned in the above table, the total number of nets that have DRV violations is 9. The count has been reduced to 3.

Final DRV Summary:

DRVs	Signal nets		Clock nets	
	Nr nets(terms)	Worst Vio	Nr nets(terms)	Worst Vio
max_cap	0 (0)	0.000	0 (0)	0.000
max_tran	0 (0)	0.000	3 (3)	-2.663

From this analysis, try to answer the following questions:

- What is the initial Signal Nets max_cap violation?
- What is the initial Signal Nets max_tran violation?
- Does the tool reduce the max_tran violation count?

4. Power Recovery: The power recovery must be done without impacting any timing and design rule violations.

Specify the following command for the power recovery:

Tcl command:

```
opt_signoff -power
```

Initial Power Summary:

```
Power Summary
-------------------------------------------------------------------------------------------------------
Group                Leakage(mW)      Internal(mW)       Switching(mW)        Total(mW)     (#insts)
-------------------------------------------------------------------------------------------------------
Combinational          0.00549661       0.111425       0.230215           0.347136         8020
Sequential             0.00242555      0.0478468      0.0184299          0.0687023          579
Pad                             0              0              0                  0            0
Block                           0              0              0                  0            0
Other                       0.004        2.91847     0.00393064            2.92641            3
-------------------------------------------------------------------------------------------------------
Total Power            0.0119222        3.07775       0.252575            3.34224         8602
-------------------------------------------------------------------------------------------------------
```

Final Power Summary:

```
Power Summary
-------------------------------------------------------------------------------------------------------
Group                Leakage(mW)      Internal(mW)       Switching(mW)        Total(mW)     (#insts)
-------------------------------------------------------------------------------------------------------
Combinational          0.00475495       0.0975458       0.223343           0.325644         7789
Sequential             0.00242555       0.0476472      0.0200119          0.0700846          579
Pad                             0              0              0                  0            0
Block                           0              0              0                  0            0
Other                       0.004        2.91847     0.00392839             2.9264            3
-------------------------------------------------------------------------------------------------------
Total Power (final)    0.0111805        3.06367       0.247283            3.32213         8371
-------------------------------------------------------------------------------------------------------
```

5. Area Recovery: The area must be reduced without impacting the setup or hold timing, and without creating any design rule violations. Area recovery is performed by downsizing both the combination and sequential cells, in addition to the buffer removal.

Specify the following command for the area recovery:

Tcl command:

```
opt_signoff -area
```

```
*info:
*info:    Total 33 instances resized or VT-Swapped
*info:
*summary:     33 cell type changes
*       :     31 instances changed cell type from 'BUF_X16' to 'BUF_X8'
*       :      1 instance  changed cell type from 'BUF_X8' to 'BUF_X4'
*       :      1 instance  changed cell type from 'INV_X32' to 'INV_X16'
*info:       total area recovered=17.8
```

LAB 1.3: Restored Analysis Flow Using Physical Master and Physical Client

Step 1. Design loading in Paradime

The design data is restored in Paradime from the physical STA saved session. In this flow, Tempus clients are not required to perform full timing updates since timing will come from the saved session for each active view. The same number of clients is required as the number of sessions to be restored. Each saved session can have data for one MMMC view. In this flow, the master is loaded with the physical data.

A snippet of Tcl scripts launched for the timing and Paradime session is given below.

- `cd Paradime_RAK/common_UI/work/`

How to run:

```
tempus -nowin -init ../scripts/sta_view1_physical.tcl.cui -
common_ui
```

- `cat ../scripts/sta_view1_physical.tcl.cui`

```
set_multi_cpu_usage -local_cpu 16
read_physical -lefs
{../../libs/lef/FreePDK45_lib_v1.0.lef ../../libs/MACRO/LEF/pllclk.le
f
../../libs/MACRO/LEF/ram_256X16A.lef ../../libs/MACRO/LEF/rom_512x16A
.lef}
read_mmmc ../work/STYLUS_DB/viewDefinition_view1.tcl
read_netlist
"../../design/ECO_INIT_11_optSetup.enc.dat/dtmf_recvr_core.v.gz" -top
dtmf_recvr_core
init_design
read_def "../../design/full.def"
read_spef -rc_corner corner_worst_RCMAX
"../../design/SPEF/corner_worst_RCMAX.spef.gz"
write_db view1_physical_cui
```

How to run:

```
tempus -nowin -init ../scripts/sta_view2_physical.tcl.cui -
common_ui
```

- `cat ../scripts/sta_view2_physical.tcl.cui`

```
set_multi_cpu_usage -local_cpu 16
read_physical -lefs
{../../libs/lef/FreePDK45_lib_v1.0.lef ../../libs/MACRO/LEF/pllclk.l
ef ../../libs/MACRO/LEF/ram_256X16A.lef ../../libs/MACRO/LEF/rom_512
x16A.lef}
read_mmmc ../work/STYLUS_DB/viewDefinition_view2.tcl
read_netlist
"../../design/ECO_INIT_11_optSetup.enc.dat/dtmf_recvr_core.v.gz"
-top dtmf_recvr_core
init_design
read_def "../../design/full.def"
read_spef -rc_corner corner_worst_RCMIN
"../../design/SPEF/corner_worst_RCMIN.spef.gz"
write_db view2_physical_cui
```

How to run:

```
tempus -nowin -init ../scripts/paradime_restoreFlow_2.tcl.cui
-eco -common_ui
```

- `cat ../scripts/paradime_restoreFlow_2.tcl.cui`

```
set_distributed_hosts -local -time_out 60
set_multi_cpu_usage -local_cpu 4 -remote_host 2 -cpu_per_remote_host
2

set_distributed_mmmc_mode -eco_merge_mmmc -eco true
set_db eco_setup_view_list {func_slow_RCMAX func_slow_RCMIN}
set_db eco_hold_view_list {func_slow_RCMAX func_slow_RCMIN}
set_db eco_def_file_list "../../design/full.def"
set_db eco_lef_file_list
"../../libs/lef/FreePDK45_lib_v1.0.lef ../../libs/MACRO/LEF/pllclk.le
f ../../libs/MACRO/LEF/ram_256X16A.lef ../../libs/MACRO/LEF/rom_512x1
6A.lef"

distribute_read_db -read_db {view1_physical_cui view2_physical_cui} -
out_dir restore_mode3_cui

set_db opt_signoff_retime path_slew_propagation
set_db opt_signoff_allow_multiple_incremental true
set_db opt_signoff_verbose true
opt_signoff -setup
opt_signoff -hold
opt_signoff -drv
opt_signoff -power
opt signoff -area
```

Step 2: Tempus ECO Optimization

After the design loading and analysis (relevant only in the full analysis flow) is done on clients, the interactive prompt is available.

Any prerequisite commands/setting for ECO run should be applied by this stage.
The ECO file written by optimization is sent automatically to clients for post-ECO signoff timing report generation.

On completion of the Tempus ECO run, the master will return to the interactive prompt with access to Tempus clients for signoff timing report generation and manual what-if analysis. You can also do successive Tempus ECO runs.

Fixing timing violations:

1. Hold Violations: Hold timing violations must be fixed without impacting the setup timing and without creating any design rule violations.

Specify the following command to fix hold violations:

Tcl command:

```
opt_signoff -hold
```

You can check the hold violations in the **Initial Hold Summary** section of the log file:

```
----------------------------------------
          Initial Hold Summary
----------------------------------------

+--------------------+-----------+
|     Hold mode      |    all    |
+--------------------+-----------+
|           WNS (ns):|    -2.201 |
|           TNS (ns):|   -59.085 |
|    Violating Paths:|        35 |
+--------------------+-----------+
```

You can check the total number of paths fixed by Tempus ECO in the **Final Hold Summary** section of the log file (the **Violating Paths** count has reduced to **0** now):

```
----------------------------------------
          Final Hold Summary
----------------------------------------

+--------------------+-----------+
|     Hold mode      |    all    |
+--------------------+-----------+
|           WNS (ns):|     0.004 |
|           TNS (ns):|     0.000 |
|    Violating Paths:|         0 |
+--------------------+-----------+
```

From this analysis, try to answer the following questions:

- Are there hold timing violations initially?
- Have all violations been fixed?

2. Setup Violations: Setup timing violations must be fixed without impacting the hold timing and without creating any design rule violations.

Specify the following command to fix setup violations:

Tcl command:

```
opt_signoff -setup
```

```
----------------------------------------
         Initial Setup Summary
----------------------------------------

+--------------------+-----------+
|     Setup mode     |    all    |
+--------------------+-----------+
|           WNS (ns):|     1.030 |
|           TNS (ns):|     0.000 |
|    Violating Paths:|         0 |
+--------------------+-----------+
```

Setup fixing is not required as there are no setup violations.

3. DRV Violations: DRV timing violations must be fixed without impacting the hold timing and setup timing violations.

Specify the following command to fix DRV violations:

Tcl command:

```
opt_signoff -drv
```

```
+----------------+-------------------------------+-----------------------------+
|                |          Signal nets          |          Clock nets         |
|    DRVs        +-----------------+-------------+---------------+-------------+
|                |  Nr nets(terms) | Worst Vio   | Nr nets(terms) | Worst Vio  |
+----------------+-----------------+-------------+---------------+-------------+
|   max_cap      |      1 (      1) |     -0.001 |     0 (     0) |      0.000 |
|   max_tran     |      5 (      5) |     -0.112 |     3 (     3) |     -2.663 |
+----------------+-------------------------------+---------------+-------------+
```

As mentioned in the above table, the total number of nets having DRV violations is 9. The count has been reduced to 6.

Final DRV Summary:

```
+----------------+-------------------------------+-----------------------------+
|                |          Signal nets          |          Clock nets         |
|    DRVs        +-----------------+-------------+---------------+-------------+
|                |  Nr nets(terms) | Worst Vio   | Nr nets(terms) | Worst Vio  |
+----------------+-----------------+-------------+---------------+-------------+
|   max_cap      |      0 (      0) |      0.000 |     0 (     0) |      0.000 |
|   max_tran     |      3 (      3) |     -0.112 |     3 (     3) |     -2.663 |
+----------------+-------------------------------+---------------+-------------+
```

From this analysis, try to answer the following questions:

- What is the initial Signal Nets max_cap violation?
- What is the initial Signal Nets max_tran violation?
- Does the tool reduce the max_tran violation count?

4. Power Recovery: The power recovery must be done without impacting any timing and design rule violations.

Specify the following command for the power recovery:

Tcl command:

```
opt_signoff -power
```

Initial Power Summary:

```
Power Summary
-------------------------------------------------------------------------------------------------------
Group                Leakage(mW)      Internal(mW)       Switching(mW)        Total(mW)     (#insts)
-------------------------------------------------------------------------------------------------------
Combinational          0.00544331       0.110704       0.224082           0.340229         7927
Sequential             0.00242555      0.0478468      0.0184299          0.0687023          579
Pad                             0              0              0                  0            0
Block                  0.00400015        2.91847      0.0104232             2.9329            4
-------------------------------------------------------------------------------------------------------
Total Power             0.011869        3.07703       0.252935            3.34183         8510
-------------------------------------------------------------------------------------------------------
```

Final Power Summary:

```
Power Summary
-------------------------------------------------------------------------------------------------------
Group                Leakage(mW)      Internal(mW)       Switching(mW)        Total(mW)     (#insts)
-------------------------------------------------------------------------------------------------------
Combinational           0.0047096      0.0970423       0.217012           0.318763         7700
Sequential             0.00242555      0.0478357      0.0200939          0.0703551          579
Pad                             0              0              0                  0            0
Block                  0.00400015        2.91847      0.0104106            2.93289            4
-------------------------------------------------------------------------------------------------------
Total Power (final)     0.0111353        3.06335       0.247516             3.322         8283
-------------------------------------------------------------------------------------------------------
```

5. Area Recovery: The area must be reduced without impacting the setup timing, hold timing, and without creating any design rule violations. Area recovery is performed by downsizing both the combination and sequential cells, in addition to the buffer removal.

Specify the following command for the area recovery:

Tcl command:

```
opt_signoff -area
```

```
*info:
*info:    Total 30 instances resized or VT-Swapped
*info:
*summary:     30 cell type changes
*       :     28 instances changed cell type from 'BUF_X16' to 'BUF_X8'
*       :      1 instance  changed cell type from 'BUF_X32' to 'BUF_X16'
*       :      1 instance  changed cell type from 'BUF_X8' to 'BUF_X4'
*info:       total area recovered=16.2
```

LAB 2: Full Analysis Flow

Overview: To perform a full analysis, specify the setup shown below similar to that used in the DMMMC setup. You may refer to the User Guide chapter on DMMMC in *Tempus documentation*.

The MMMC design loading setup to load the design and the Batch mode script is done using the `distribute_design_views` command.

```
distribute_design_views -design_script
../scripts/sta_view.tcl.cui -out_dir scratch_mode1_cui -views
[list func_slow_RCMAX func_slow_RCMIN] -analysis_script
../scripts/sta.tcl.cui -write_db all
```

The master launches multiple Tempus DMMMC clients to load the design data (as specified in `sta_view.tcl.cui`) and execute the contents of `sta.tcl.cui` (to perform full TA runs, generate timing reports, and so on). Meanwhile, the master also loads the MMMC design for all views that are loaded on clients.

Lab Flow

The flow consists of the following steps:

- Full Analysis Flow Using Logical Master and Logical Client
- Tempus ECO Optimization
- Full Analysis Flow Using Physical Master and Physical Client
- Tempus ECO Optimization

LAB 2.1: Full Analysis Flow Using Logical Master and Logical Client

Step 1. Design loading in Paradime

The design data is loaded in Paradime from the logical STA script. The same number of clients is required as the number of views to be launched.

A snippet of Tcl scripts launched for the timing and Paradime session is given below.

- `cd Paradime_RAK/common_UI/work`
- `cat ../scripts/sta_view.tcl.cui`

```
read_mmmc ../work/STYLUS_DB/viewDefinition.tcl
read_netlist
"../../design/ECO_INIT_11_optSetup.enc.dat/dtmf_recvr_core.v.gz" -
top dtmf_recvr_core
init_design
```

- cat ../scripts/sta.tcl.cui

```
source ../scripts/spef.tcl
set_db delaycal_enable_si true
```

How to run:

```
tempus -nowin -init
../scripts/paradime_logical_scratch.tcl.cui -eco -common_ui
```

- cat ../scripts/paradime_logical_scratch.tcl.cui

```
set_distributed_hosts -local -time_out 60
set_multi_cpu_usage -local_cpu 4 -remote_host 2 -
cpu_per_remote_host 2

set_distributed_mmmc_mode -eco_merge_mmmc -eco true
set_db eco_setup_view_list {func_slow_RCMAX func_slow_RCMIN}
set_db eco_hold_view_list {func_slow_RCMAX func_slow_RCMIN}

distribute_design_views -design_script ../scripts/sta_view.tcl.cui
-out_dir scratch_mode1_cui -views [list func_slow_RCMAX
func_slow_RCMIN] -analysis_script ../scripts/sta.tcl.cui -write_db
all

set_db opt_signoff_retime path_slew_propagation
set_db opt_signoff_allow_multiple_incremental true
set_db opt_signoff_verbose true
opt_signoff -setup
opt_signoff -hold
opt_signoff -drv
opt_signoff -power
opt_signoff -area
```

On completion of the Tempus ECO run, the master will return to the interactive prompt with access to Tempus clients for signoff timing report generation and manual what-if analysis. You can also do successive Tempus ECO runs.

The settings given below are applied for optimization.

Fixing timing violations:

1. Hold Violations: Hold timing violations must be fixed without impacting the setup timing and without creating any design rule violations.

Specify the following command to fix hold violations:

Tcl command:

```
opt_signoff -hold
```

You can check the hold violations in the **Initial Hold Summary** section of the log file:

```
------------------------------------
          Initial Hold Summary
------------------------------------

+--------------------+-----------+
|     Hold mode      |    all    |
+--------------------+-----------+
|           WNS (ns):|    -2.232 |
|           TNS (ns):|   -59.791 |
|    Violating Paths:|        35 |
+--------------------+-----------+
```

You can check the total number of paths fixed by Tempus ECO in the **Final Hold Summary** of the log file (the **Violating Paths** count has reduced to **0** now):

```
------------------------------------
          Final Hold Summary
------------------------------------

+--------------------+-----------+
|     Hold mode      |    all    |
+--------------------+-----------+
|           WNS (ns):|     0.005 |
|           TNS (ns):|     0.000 |
|    Violating Paths:|         0 |
+--------------------+-----------+
```

From this analysis, try to answer the following questions:

- Are there hold timing violations initially?
- Have all violations been fixed?

2. Setup Violations: Setup timing violations must be fixed without impacting the hold timing and without creating any design rule violations.

Specify the following command to fix setup violations:

Tcl command:

```
opt_signoff -setup
```

```
----------------------------------------
         Initial Setup Summary
----------------------------------------

+--------------------+-----------+
|     Setup mode     |    all    |
+--------------------+-----------+
|           WNS (ns):|     0.403 |
|           TNS (ns):|     0.000 |
|    Violating Paths:|         0 |
+--------------------+-----------+
```

Setup fixing is not required as there are no setup violations.

3. DRV violations: DRV timing violations must be fixed without impacting the hold timing and setup timing violations.

Specify the following command to fix DRV violations:

Tcl command:

```
opt_signoff -drv
```

Initial DRV Summary:

DRVs	Signal nets		Clock nets	
	Nr nets(terms)	Worst Vio	Nr nets(terms)	Worst Vio
max_cap	1 (1)	-0.001	0 (0)	0.000
max_tran	14 (14)	-0.466	3 (3)	-2.663

As mentioned in the above table, the total number of nets having DRV violations is 18. The count has been reduced to 3.

Final DRV Summary:

DRVs	Signal nets		Clock nets	
	Nr nets(terms)	Worst Vio	Nr nets(terms)	Worst Vio
max_cap	0 (0)	0.000	0 (0)	0.000
max_tran	0 (0)	0.000	3 (3)	-2.663

From this analysis, try to answer the following questions:

- What is the initial Signal Nets max_cap violation?
- What is the initial Signal Nets max_tran violation?
- Does the tool reduce the max_tran violation count?

4. Power Recovery: The power recovery must be without impacting any timing and design rule violations.

Specify the following command for the power recovery:

Tcl command:

```
opt_signoff -power
```

Initial Power Summary:

```
Power Summary
-------------------------------------------------------------------------------------------------------
Group                Leakage(mW)      Internal(mW)       Switching(mW)        Total(mW)     (#insts)
-------------------------------------------------------------------------------------------------------
Combinational          0.00550966       0.111692       0.230242           0.347444         8024
Sequential             0.00242555      0.0478465      0.0184299           0.068702          579
Pad                             0              0              0                  0            0
Block                           0              0              0                  0            0
Other                       0.004        2.91847     0.00393131            2.92641            3
-------------------------------------------------------------------------------------------------------
Total Power            0.0119352        3.07801       0.252603           3.34255         8606
-------------------------------------------------------------------------------------------------------
```

Final Power Summary:

```
Power Summary
-------------------------------------------------------------------------------------------------------
Group                Leakage(mW)      Internal(mW)       Switching(mW)        Total(mW)     (#insts)
-------------------------------------------------------------------------------------------------------
Combinational          0.00477552       0.0978229       0.223396           0.325994         7798
Sequential             0.00242555      0.0476469      0.0200106          0.0700831          579
Pad                             0              0              0                  0            0
Block                           0              0              0                  0            0
Other                       0.004        2.91847     0.00393631            2.92641            3
-------------------------------------------------------------------------------------------------------
Total Power (final)    0.0112011        3.06394       0.247343           3.32249         8380
-------------------------------------------------------------------------------------------------------
```

5. Area Recovery: The area must be reduced without impacting the setup timing, hold timing, and without creating any design rule violations. Area recovery is performed by downsizing both the combination and sequential cells, in addition to the buffer removal.

Specify the following command to achieve area recovery:

Tcl command:

```
opt_signoff -area
```

```
*info:
*info:    Total 33 instances resized or VT-Swapped
*info:
*summary:     34 cell type changes
*       :      1 instance  changed cell type from 'BUF_X16' to 'BUF_X1'
*       :      1 instance  changed cell type from 'BUF_X16' to 'BUF_X4'
*       :     29 instances changed cell type from 'BUF_X16' to 'BUF_X8'
*       :      2 instances changed cell type from 'BUF_X32' to 'BUF_X16'
*       :      1 instance  changed cell type from 'BUF_X8' to 'BUF_X4'
*info:       total area recovered=19.4
```

LAB 2.2: Full Analysis Flow Using Physical Master and Physical Client

Step 1. Design loading in Paradime

The design data is loaded in Paradime from the physical STA script. The same number of clients is required as the number of views to be launched. In this flow, the master is also loaded with the physical data.

A snippet of Tcl scripts launched for the timing and Paradime session is given below.

- `cd Paradime_RAK/common_UI/work`
- `cat ../scripts/sta_view_physical.tcl.cui`

```
read_physical -lefs
{../../libs/lef/FreePDK45_lib_v1.0.lef ../../libs/MACRO/LEF/pllclk.l
ef ../../libs/MACRO/LEF/ram_256X16A.lef ../../libs/MACRO/LEF/rom_512
x16A.lef}
read_mmmc ../work/STYLUS_DB/viewDefinition.tcl
read_netlist
"../../design/ECO_INIT_11_optSetup.enc.dat/dtmf_recvr_core.v.gz" -
top dtmf_recvr_core
init_design
read_def "../../design/full.def"
```

- `cat ../scripts/sta.tcl.cui`

```
source ../scripts/spef.tcl
set_db delaycal_enable_si true
```

How to run:

```
tempus -nowin -init
../scripts/paradime_physical_scratch.tcl.cui -eco -common_ui
```

- `cat ../scripts/paradime_physical_scratch.tcl.cui`

```
set_distributed_hosts -local -time_out 60
set_multi_cpu_usage -local_cpu 4 -remote_host 2 -
cpu_per_remote_host 2

set_distributed_mmmc_mode -eco_merge_mmmc -eco true
set_db eco_setup_view_list {func_slow_RCMAX func_slow_RCMIN}
set_db eco_hold_view_list {func_slow_RCMAX func_slow_RCMIN}
set_db eco_def_file_list "../../design/full.def"
set_db eco_lef_file_list
"../../libs/lef/FreePDK45_lib_v1.0.lef ../../libs/MACRO/LEF/pllclk.
lef
../../libs/MACRO/LEF/ram_256X16A.lef ../../libs/MACRO/LEF/rom_512x1
6A.lef"

distribute_design_views -
design_script ../scripts/sta_view_physical.tcl.cui -out_dir
scratch_mode2_cui -views [list func_slow_RCMAX func_slow_RCMIN] -
analysis_script ../scripts/sta.tcl.cui -write_db all

set_db opt_signoff_retime path_slew_propagation
set_db opt_signoff_allow_multiple_incremental true
set_db opt_signoff_verbose true
opt_signoff -setup
opt_signoff -hold
opt_signoff -drv
opt_signoff -power
opt_signoff -area
```

Fixing timing violations:

1. Hold Violations: Hold timing violations must be fixed without impacting the setup timing and without creating any design rule violations.

Specify the following command to fix hold violations:

Tcl command:

```
opt_signoff -hold
```

```
------------------------------------
         Initial Hold Summary
------------------------------------

+--------------------+-----------+
|     Hold mode      |    all    |
+--------------------+-----------+
|           WNS (ns):|    -2.232 |
|           TNS (ns):|   -59.791 |
|    Violating Paths:|        35 |
+--------------------+-----------+
```

You can check the total number of paths fixed by Tempus ECO in the **Final Hold Summary** section of the log file (the **Violating Paths** count has reduced to **0** now):

```
------------------------------------
          Final Hold Summary
------------------------------------

+--------------------+-----------+
|     Hold mode      |    all    |
+--------------------+-----------+
|           WNS (ns):|     0.001 |
|           TNS (ns):|     0.000 |
|    Violating Paths:|         0 |
+--------------------+-----------+
```

From this analysis, try to answer the following questions:

- Are there hold timing violations initially?
- Have all violations been fixed?

2. Setup Violations: Setup timing violations must be fixed without impacting the hold timing and without creating any design rule violations.

Specify the following command to fix setup violations:

Tcl command:

```
opt_signoff -setup
```

```
------------------------------------
         Initial Setup Summary
------------------------------------

+--------------------+-----------+
|     Setup mode     |    all    |
+--------------------+-----------+
|           WNS (ns):|     0.403 |
|           TNS (ns):|     0.000 |
|    Violating Paths:|         0 |
+--------------------+-----------+
```

Setup fixing is not required as there are no setup violations.

3. DRV Violations: DRV timing violations must be fixed without impacting the hold timing and setup timing violations.

Specify the following command to fix DRV violations:

Tcl command:

```
opt_signoff -drv
```

Initial DRV Summary:

```
+----------------+-------------------------------+-----------------------------+
|                |          Signal nets          |          Clock nets         |
|    DRVs        +------------------+------------+----------------+------------+
|                |  Nr nets(terms)  | Worst Vio  | Nr nets(terms) | Worst Vio  |
+----------------+------------------+------------+----------------+------------+
|   max_cap      |      1 (      1) |     -0.001 |     0 (     0) |      0.000 |
|   max_tran     |     14 (     14) |     -0.466 |     3 (     3) |     -2.663 |
+----------------+------------------+------------+----------------+------------+
```

As mentioned in the above table, the total number of nets having DRV violations is 18. The count has been reduced to 7.

Final DRV Summary:

DRVs	Signal nets		Clock nets	
	Nr nets(terms)	Worst Vio	Nr nets(terms)	Worst Vio
max_cap	0 (0)	0.000	0 (0)	0.000
max_tran	4 (4)	-0.466	3 (3)	-2.663

From this analysis, try to answer the following questions:

- What is the initial Signal Nets max_cap violation?
- What is the initial Signal Nets max_tran violation?
- Does the tool reduce the max_tran violation count?

4. Power Recovery: The power recovery must be done without impacting any timing and design rule violations.

Specify the following command for the power recovery:

Tcl command:

```
opt_signoff -power
```

Initial Power Summary:

Power Summary

Group	Leakage(mW)	Internal(mW)	Switching(mW)	Total(mW)	(#insts)
Combinational	0.00545601	0.11096	0.224139	0.340555	7934
Sequential	0.00242555	0.0478465	0.0184299	0.068702	579
Pad	0	0	0	0	0
Block	0.00400015	2.91847	0.0104254	2.9329	4
Total Power	0.0118817	3.07728	0.252995	3.34216	8517

Final Power Summary:

Power Summary

Group	Leakage(mW)	Internal(mW)	Switching(mW)	Total(mW)	(#insts)
Combinational	0.00472792	0.0972893	0.217092	0.31911	7719
Sequential	0.00242555	0.0478348	0.0201027	0.0703631	579
Pad	0	0	0	0	0
Block	0.00400015	2.91847	0.0104149	2.93289	4
Total Power (final)	0.0111536	3.0636	0.24761	3.32236	8302

5. Area Recovery: The area must be reduced without impacting the setup timing, hold timing, and without creating any design rule violations. Area recovery is performed by downsizing both the combination and sequential cells, in addition to the buffer removal.

Specify the following command for the area recovery:

Tcl command:

```
opt_signoff -area
```

```
*info:
*info:    Total 29 instances resized or VT-Swapped
*info:
*summary:     29 cell type changes
*       :      1 instance  changed cell type from 'BUF_X16' to 'BUF_X1'
*       :     25 instances changed cell type from 'BUF_X16' to 'BUF_X8'
*       :      2 instances changed cell type from 'BUF_X32' to 'BUF_X16'
*       :      1 instance  changed cell type from 'BUF_X8' to 'BUF_X4'
*info:       total area recovered=16.5
```

Support

Cadence Support Portal provides access to support resources, including an extensive knowledge base, access to software updates for Cadence products, and the ability to interact with Cadence Customer Support. Visit https://support.cadence.com.

Feedback

Email comments, questions, and suggestions to content_feedback@cadence.com.