**Cadance.sh**

#!/bin/bash

export MMSIM\_ROOT=/eda/cadence/mmsim121

#export OA\_HOME=/eda/cadence/ic615/oa\_v22.41.022/

#original export OA\_HOME=/eda/cadence/ic616\_005/oa\_v22.43.018/

export OA\_HOME=/eda/cadence/INNOVUS171/oa\_v22.50.063

#export OA\_HOME=/eda/cadence/ic614/oa\_v22.04.072/

#export CDS\_ROOT=/eda/cadence/ic615

export CDS\_ROOT=/eda/cadence/ic616\_005

#export CDS\_ROOT=/eda/cadence/ic614

export X\_DIR=/process/xfab

export PATH=$X\_DIR/x\_all/cadence/xenv:$PATH

export CDSDIR=$CDS\_ROOT

export CDSHOME=$CDS\_ROOT

export CDS\_ROOT=$CDS\_ROOT

export CDS\_INST\_DIR=$CDS\_ROOT

export CDS\_Netlisting\_Mode=Analog

#export CDS\_Netlisting\_Mode Digital

export PATH=$CDS\_INST\_DIR/tools/bin:$PATH

export PATH=$CDS\_INST\_DIR/tools/dfII/bin:$PATH

export PATH=$CDS\_INST\_DIR/tools/plot/bin:$PATH

export PATH=$CDS\_INST\_DIR/tools/dracula/bin:$PATH

export PATH=$CDS\_ROOT/tools/bin:$PATH

export PATH=$CDS\_ROOT/tools/dfII/bin:$PATH

export PATH=$CDS\_ROOT/tools/dracula/bin:$PATH

export PATH=$CDS\_ROOT/tools/plot/bin:$PATH

export PATH=$CDS\_ROOT/tools/iccraft/bin:$PATH

export PATH=$MMSIM\_ROOT/tools/dfII/bin:$PATH

export PATH=$MMSIM\_ROOT/tools/spectre/bin:$PATH

export PATH=$MMSIM\_ROOT/tools/ultrasim/bin:$PATH

export PATH=$MMSIM\_ROOT/tools/bin:$PATH

export LM\_LICENSE\_FILE=/usr/license/license\_new.cds

#export LM\_LICENSE\_FILE=5280@BICASL1:27001@BICASL1

export CDS\_LOAD\_ENV=CSF

#export LD\_LIBRARY\_PATH=/lib:/lib64:/usr/lib:/usr/lib64:$LD\_LIBRARY\_PATH

export HSPICE\_HOME=/eda/synopsys/hspice/1303

export PATH=$HSPICE\_HOME/hspice/amd64:$PATH

#export CDS\_LOAD\_ENV= CSF

export ASSURAHOME=/eda/cadence/ASSURA616

export PATH=$ASSURAHOME/tools/assrua/bin:$ASSURAHOME/tools/bin:$PATH

export IUS\_PATH=/eda/cadence/incisive13

#export IUS\_PATH=/eda/cadence/ius92

#export CDS\_ROOT=$IUS\_PATH

export CDS\_INST\_DIR=$IUS\_PATH

export CDS\_ROOT=/eda/cadence/incisive13

export PATH=$IUS\_PATH/tools/bin:$PATH

export LD\_LIBRARY\_PATH=$IUS\_PATH/tools/systemc/gcc/install/lib:$IUS\_PATH/tools/lib:$IUS\_PATH/tools/verilog/lib:$LD\_LIBRARY\_PATH

#export XFAB\_CALIBRE\_RUNSET=/process/xfab/xh035/mentor/v6\_3/calibre/v6\_3\_2

#export XFAB\_CALIBRE\_RUNSET=/process/xfab/xh035/mentor/v6\_2/calibre/v6\_2\_4

#export XFAB\_CALIBRE\_RUNSET=/process/xfab/xh035/mentor/v6\_1/calibre/v6\_1\_8

#export XFAB\_PEX\_RUNSET=/process/xfab/xh035/mentor/v6\_0/PEX\_calibre/v6\_0\_1\_2

#export XFAB\_PEX\_RUNSET=/process/xfab/xh035/mentor/v6\_2/PEX\_calibre/v6\_2\_1

#source /eda/env/menter2015.bashrc

#source /eda/env/menter2014.cshrc

export CDS\_AUTO\_64BIT=ALL

export PATH=/eda/cadence/INNOVUS171/bin:$PATH

**makefile**

VERSION=17.10-p003\_1

VPATH=./make

TOOL=innovus

SCRIPTS=FF

LOG=LOG

ARGS=-64 -nowin

PARALLEL=-j2

SUBMIT=""

TARGET=signoff

FF\_STOP=signoff

STEPS = version setup assemble partition\_place assign\_pin partition do\_cleanup

TOP=dtmf\_recvr\_core

all: setup partition\_place assign\_pin partition blocks.$(TARGET) top.$(TARGET) assemble

help:

@echo "==================================================="

@echo " $(VERSION) Foundation Flows"

@echo "==================================================="

@echo " Makefile Targets"

@echo "==================================================="

@echo " all : Run complete flow (default)"

@echo " partition\_place : Initial placement & feedthrough"

@echo " assign\_pin : Pin assignment"

@echo " partition : Partition design"

@echo " blocks.<target> : Implement blocks"

@echo " assemble : Assemble design"

@echo "==================================================="

@echo " Makefile Options"

@echo "==================================================="

@echo " VPATH : Make directory (default make)"

@echo " SUBMIT : LSF launch command (default '')"

@echo " PARALLEL : Number of machines (default -j2)"

@echo " TOOL : INNOVUS executable (default innovus)"

@echo " ARGS : INNOVUS arguments (default -nowin -64)"

@echo " SCRIPTS : Script directory (default FF)"

@echo " LOG : Logfile directory (default LOG)"

@echo "==================================================="

assemble: top.$(TARGET)

VPATH=$(VPATH); export VPATH; $(TOOL) $(ARGS) -init $(SCRIPTS)/INNOVUS/run\_assemble.tcl -log $(LOG)/assemble.log -overwrite

/bin/touch $(VPATH)/assemble

top.$(TARGET) : blocks.$(TARGET)

cd PARTITION/$(TOP); $(MAKE) -f Makefile FF\_STOP=$(FF\_STOP) $(TARGET)

/bin/touch $(VPATH)/$@

blocks.$(TARGET) : partition

cd PARTITION; VPATH=$(VPATH); export VPATH; $(MAKE) $(PARALLEL) blocks FF\_STOP=$(FF\_STOP) TARGET=$(TARGET)

/bin/touch $(VPATH)/$@

partition\_place : setup

VPATH=$(VPATH); export VPATH; $(TOOL) $(ARGS) -init $(SCRIPTS)/INNOVUS/run\_partition\_place.tcl -log $(LOG)/partition\_place.log -overwrite

/bin/touch $(VPATH)/partition\_place

assign\_pin : partition\_place

VPATH=$(VPATH); export VPATH; $(TOOL) $(ARGS) -init $(SCRIPTS)/INNOVUS/run\_assign\_pin.tcl -log $(LOG)/assign\_pin.log -overwrite

/bin/touch $(VPATH)/assign\_pin

partition : assign\_pin

VPATH=$(VPATH); export VPATH; $(TOOL) $(ARGS) -init $(SCRIPTS)/INNOVUS/run\_partition.tcl -log $(LOG)/partition.log -overwrite

/bin/touch $(VPATH)/partition

single:

@$(MAKE) TARGET=single FF\_STOP=$(FF\_STOP)

debug\_%:

export STEP=$\* ; VPATH=$(VPATH); export VPATH; $(TOOL) -init $(SCRIPTS)/INNOVUS/run\_debug.tcl -log $(LOG)/$@.log -win $(ARGS:-nowin=)

reset:

@/bin/rm -f $(VPATH)/\* PARTITION/\*/$(VPATH)/\* PARTITION/\*/$(VPATH)/.RUNNING

setup:

# /bin/rm -fr PARTITION $(VPATH) LOG

/bin/mkdir -p $(VPATH) LOG

/bin/touch $(VPATH)/setup

**Setup.tcl**

global vars

set vars(version) 17.1.0

#######################################################################

# Define some variables to point to data, libraries, and scripts

#----------------------------------------------------------------------

#set vars(design\_root) ""

set vars(script\_root) ./SCRIPTS

set vars(dbs\_dir) ./DBS

set vars(rpt\_dir) ./RPT

#######################################################################

# Define technology and physical libraries

#-----------------------------------------------------------------

set vars(process) 45nm

set vars(lef\_files) \

"/home/yongfu/proj/jizuo/aeda/eda/LEF/gsclib045\_v3.5/lef/gsclib045\_tech.lef \

/home/yongfu/proj/jizuo/aeda/eda/LEF/gsclib045\_v3.5/lef/gsclib045\_macro.lef \

/home/yongfu/proj/jizuo/aeda/eda/LEF/macro\_lefs/CDK\_R512x16.lef \

/home/yongfu/proj/jizuo/aeda/eda/LEF/macro\_lefs/CDK\_S256x16.lef \

/home/yongfu/proj/jizuo/aeda/eda/LEF/macro\_lefs/pllclk.lef"

#set vars(ilm\_list) ""

set vars(partition\_list) "GROUP1 GROUP2 GROUP3 GROUP4 GROUP5 GROUP6"

#######################################################################

# Define the design data

#-----------------------------------------------------------------

set vars(netlist) "/home/yongfu/proj/jizuo/aeda/eda/work/dtmf\_recvr\_core\_1123new.v"

set vars(design) dtmf\_recvr\_core

set vars(fp\_file) "dtmf\_recvr\_core.fp"

#set vars(def\_files) ""

#set vars(fp\_tcl\_file) ""

#set vars(fp\_tcl\_proc) ""

#set vars(scan\_def) ""

set vars(power\_nets) VDD

set vars(ground\_nets) VSS

set vars(honor\_pitch) FALSE

#set vars(cts\_spec) ""

set vars(cts\_cells) "CLKAND2X12 CLKBUFX2 CLKBUFX8 CLKBUFX12 CLKINVX3 CLKINVX6 CLKINVX16 CLKAND2X4 CLKBUFX3 CLKBUFX6 CLKBUFX16 CLKINVX1 CLKINVX4 CLKAND2X8 CLKINVX20 CLKAND2X2 CLKBUFX4 CLKBUFX20 CLKINVX2 CLKINVX8 CLKINVX12 CLKAND2X6 CLKAND2X3"

set vars(enable\_ss) false

set vars(flow) mmmc

set vars(enable\_ocv) pre\_postcts

#######################################################################

# Define library sets ...

#-----------------------------------------------------------------

set vars(library\_sets) "dtmf\_libs\_min dtmf\_libs\_max dtmf\_libs\_typical"

#set vars(dtmf\_libs\_min,si) ""

set vars(dtmf\_libs\_min,timing) \

"/home/yongfu/proj/jizuo/aeda/eda/LIB/gsclib045\_v3.5/timing/fast.lib \

/home/yongfu/proj/jizuo/aeda/eda/LIB/macro\_libs/CDK\_R512x16.lib \

/home/yongfu/proj/jizuo/aeda/eda/LIB/macro\_libs/CDK\_S128x16.lib \

/home/yongfu/proj/jizuo/aeda/eda/LIB/macro\_libs/CDK\_S256x16.lib \

/home/yongfu/proj/jizuo/aeda/eda/LIB/macro\_libs/pllclk\_slow.lib"

#set vars(dtmf\_libs\_max,si) ""

set vars(dtmf\_libs\_max,timing) \

"/home/yongfu/proj/jizuo/aeda/eda/LIB/gsclib045\_v3.5/timing/slow.lib \

/home/yongfu/proj/jizuo/aeda/eda/LIB/macro\_libs/CDK\_R512x16.lib \

/home/yongfu/proj/jizuo/aeda/eda/LIB/macro\_libs/CDK\_S128x16.lib \

/home/yongfu/proj/jizuo/aeda/eda/LIB/macro\_libs/CDK\_S256x16.lib \

/home/yongfu/proj/jizuo/aeda/eda/LIB/macro\_libs/pllclk\_slow.lib"

#set vars(dtmf\_libs\_typical,si) ""

set vars(dtmf\_libs\_typical,timing) \

"/home/yongfu/proj/jizuo/aeda/eda/LIB/gsclib045\_v3.5/timing/typical.lib \

/home/yongfu/proj/jizuo/aeda/eda/LIB/macro\_libs/CDK\_R512x16.lib \

/home/yongfu/proj/jizuo/aeda/eda/LIB/macro\_libs/CDK\_S128x16.lib \

/home/yongfu/proj/jizuo/aeda/eda/LIB/macro\_libs/CDK\_S256x16.lib \

/home/yongfu/proj/jizuo/aeda/eda/LIB/macro\_libs/pllclk\_slow.lib"

#######################################################################

# Define rc corners ...

#-----------------------------------------------------------------

set vars(rc\_corners) dtmf\_rc\_corner

#set vars(dtmf\_rc\_corner,qx\_lib\_file) ""

#set vars(dtmf\_rc\_corner,qx\_conf\_file) ""

#set vars(dtmf\_rc\_corner,cap\_table) ""

set vars(dtmf\_rc\_corner,qx\_tech\_file) /home/yongfu/proj/jizuo/aeda/eda/QRC/t018s6mm.tch

#set vars(dtmf\_rc\_corner,T) ""

set vars(dtmf\_rc\_corner,pre\_route\_res\_factor) 1.00

set vars(dtmf\_rc\_corner,pre\_route\_cap\_factor) 1.00

set vars(dtmf\_rc\_corner,pre\_route\_clk\_res\_factor) 0.00

set vars(dtmf\_rc\_corner,pre\_route\_clk\_cap\_factor) 0.00

set vars(dtmf\_rc\_corner,post\_route\_res\_factor) "1.00 1.00 1.00"

set vars(dtmf\_rc\_corner,post\_route\_cap\_factor) "1.00 1.00 1.00"

set vars(dtmf\_rc\_corner,post\_route\_xcap\_factor) "1.00 1.00 1.00"

set vars(dtmf\_rc\_corner,post\_route\_clk\_res\_factor) "0.00 1.00 1.00"

set vars(dtmf\_rc\_corner,post\_route\_clk\_cap\_factor) "0.00 1.00 1.00"

#set vars(dtmf\_rc\_corner,xcap\_factor) ""

#######################################################################

# Define delay corners ...

#-----------------------------------------------------------------

set vars(enable\_cppr) none

set vars(delay\_corners) "dtmf\_corner\_min dtmf\_corner\_typical dtmf\_corner\_max"

#set vars(dtmf\_corner\_min,clock\_cell\_early) ""

#set vars(dtmf\_corner\_min,clock\_cell\_late) ""

#set vars(dtmf\_corner\_min,clock\_net\_early) ""

#set vars(dtmf\_corner\_min,clock\_net\_late) ""

#set vars(dtmf\_corner\_min,data\_cell\_early) ""

#set vars(dtmf\_corner\_min,data\_cell\_late) ""

#set vars(dtmf\_corner\_min,data\_net\_early) ""

#set vars(dtmf\_corner\_min,data\_net\_late) ""

set vars(dtmf\_corner\_min,rc\_corner) dtmf\_rc\_corner

set vars(dtmf\_corner\_min,library\_set) dtmf\_libs\_min

set vars(dtmf\_corner\_min,cell\_check\_late) dtmf\_libs\_min

set vars(dtmf\_corner\_min,cell\_check\_early) dtmf\_libs\_min

#set vars(dtmf\_corner\_typical,clock\_cell\_early) ""

#set vars(dtmf\_corner\_typical,clock\_cell\_late) ""

#set vars(dtmf\_corner\_typical,clock\_net\_early) ""

#set vars(dtmf\_corner\_typical,clock\_net\_late) ""

#set vars(dtmf\_corner\_typical,data\_cell\_early) ""

#set vars(dtmf\_corner\_typical,data\_cell\_late) ""

#set vars(dtmf\_corner\_typical,data\_net\_early) ""

#set vars(dtmf\_corner\_typical,data\_net\_late) ""

set vars(dtmf\_corner\_typical,rc\_corner) dtmf\_rc\_corner

set vars(dtmf\_corner\_typical,library\_set) dtmf\_libs\_typical

set vars(dtmf\_corner\_typical,cell\_check\_late) dtmf\_libs\_typical

set vars(dtmf\_corner\_typical,cell\_check\_early) dtmf\_libs\_typical

#set vars(dtmf\_corner\_max,clock\_cell\_early) ""

#set vars(dtmf\_corner\_max,clock\_cell\_late) ""

#set vars(dtmf\_corner\_max,clock\_net\_early) ""

#set vars(dtmf\_corner\_max,clock\_net\_late) ""

#set vars(dtmf\_corner\_max,data\_cell\_early) ""

#set vars(dtmf\_corner\_max,data\_cell\_late) ""

#set vars(dtmf\_corner\_max,data\_net\_early) ""

#set vars(dtmf\_corner\_max,data\_net\_late) ""

set vars(dtmf\_corner\_max,rc\_corner) dtmf\_rc\_corner

set vars(dtmf\_corner\_max,library\_set) dtmf\_libs\_max

set vars(dtmf\_corner\_max,cell\_check\_late) dtmf\_libs\_max

set vars(dtmf\_corner\_max,cell\_check\_early) dtmf\_libs\_max

#######################################################################

# Define constraints modes ...

#-----------------------------------------------------------------

set vars(constraint\_modes) "dtmf\_constraint\_slow dtmf\_constraint\_typical dtmf\_constraint\_fast"

set vars(dtmf\_constraint\_slow,pre\_cts\_sdc) "/home/yongfu/proj/jizuo/aeda/eda/Constraints/mmmc/dtmf\_recvr\_core\_gate\_slow.sdc"

#set vars(dtmf\_constraint\_slow,incr\_cts\_sdc) ""

set vars(dtmf\_constraint\_slow,post\_cts\_sdc) "../Constraints/mmmc/dtmf\_recvr\_core\_gate\_slow.sdc"

#set vars(dtmf\_constraint\_slow,ilm\_pre\_cts\_sdc) ""

#set vars(dtmf\_constraint\_slow,ilm\_incr\_cts\_sdc) ""

#set vars(dtmf\_constraint\_slow,ilm\_post\_cts\_sdc) ""

set vars(dtmf\_constraint\_typical,pre\_cts\_sdc) "/home/yongfu/proj/jizuo/aeda/eda/Constraints/mmmc/dtmf\_recvr\_core\_gate\_typical.sdc"

#set vars(dtmf\_constraint\_typical,incr\_cts\_sdc) ""

set vars(dtmf\_constraint\_typical,post\_cts\_sdc) "../Constraints/mmmc/dtmf\_recvr\_core\_gate\_typical.sdc"

#set vars(dtmf\_constraint\_typical,ilm\_pre\_cts\_sdc) ""

#set vars(dtmf\_constraint\_typical,ilm\_incr\_cts\_sdc) ""

#set vars(dtmf\_constraint\_typical,ilm\_post\_cts\_sdc) ""

set vars(dtmf\_constraint\_fast,pre\_cts\_sdc) "/home/yongfu/proj/jizuo/aeda/eda/Constraints/mmmc/dtmf\_recvr\_core\_gate\_fast.sdc"

#set vars(dtmf\_constraint\_fast,incr\_cts\_sdc) ""

set vars(dtmf\_constraint\_fast,post\_cts\_sdc) "../Constraints/mmmc/dtmf\_recvr\_core\_gate\_fast.sdc"

#set vars(dtmf\_constraint\_fast,ilm\_pre\_cts\_sdc) ""

#set vars(dtmf\_constraint\_fast,ilm\_incr\_cts\_sdc) ""

#set vars(dtmf\_constraint\_fast,ilm\_post\_cts\_sdc) ""

#######################################################################

# Define analysis views ...

#-----------------------------------------------------------------

set vars(default\_hold\_view) dtmf\_view\_fast

set vars(default\_setup\_view) dtmf\_view\_slow

set vars(hold\_analysis\_views) dtmf\_view\_fast

set vars(dtmf\_view\_fast,constraint\_mode) dtmf\_constraint\_fast

set vars(dtmf\_view\_fast,delay\_corner) dtmf\_corner\_min

set vars(setup\_analysis\_views) dtmf\_view\_slow

set vars(dtmf\_view\_slow,constraint\_mode) dtmf\_constraint\_slow

set vars(dtmf\_view\_slow,delay\_corner) dtmf\_corner\_max

set vars(active\_hold\_views) dtmf\_view\_fast

set vars(active\_setup\_views) dtmf\_view\_slow

#######################################################################

# Define power settings ...

#-----------------------------------------------------------------

set vars(power\_analysis\_view) dtmf\_view\_slow

set vars(leakage\_power\_effort) high

set vars(dynamic\_power\_effort) high

#set vars(activity\_file) ""

set vars(activity\_file\_format) TCF

set vars(report\_power) TRUE

#set vars(cpf\_file) ""

set vars(cpf\_keep\_rows) FALSE

set vars(cpf\_power\_domain) FALSE

set vars(cpf\_power\_switch) FALSE

set vars(cpf\_isolation) FALSE

set vars(cpf\_state\_retention) FALSE

set vars(cpf\_level\_shifter) FALSE

#######################################################################

# Define tool specific options ...

#-----------------------------------------------------------------

set vars(max\_route\_layer) 6

set vars(generate\_tracks) TRUE

set vars(postroute\_extraction\_effort) high

set vars(multi\_cut\_effort) high

set vars(litho\_driven\_routing) FALSE

set vars(postroute\_spread\_wires) FALSE

#set vars(delta\_delay\_threshold) ""

#set vars(celtic\_settings) ""

#set vars(coupling\_c\_thresh) ""

#set vars(relative\_c\_thresh) ""

#set vars(total\_c\_thresh) ""

set vars(si\_analysis\_type) default

set vars(signoff\_extraction\_effort) high

#set vars(antenna\_diode) ""

set vars(metalfill) false

#set vars(metalfill\_tcl) ""

#set vars(gds\_files) ""

#set vars(qrc\_layer\_map) ""

#set vars(qrc\_library) ""

#set vars(qrc\_config\_file) ""

#set vars(gds\_layer\_map) ""

#set vars(oa\_abstract\_name) ""

#set vars(oa\_layout\_name) ""

#set vars(oa\_ref\_lib) ""

#set vars(oa\_design\_lib) ""

#set vars(oa\_design\_cell) ""

#set vars(oa\_design\_view) ""

set vars(local\_cpus) 4

#set vars(remote\_hosts) ""

#set vars(cpu\_per\_remote\_host) ""

set vars(distribute) local

#set vars(custom,script) ""

#set vars(lsf,queue) ""

#set vars(lsf,resource) ""

#set vars(lsf,args) ""

#set vars(rsh,host\_list) ""

#set vars(distribute\_timeout) ""

**Run\_partition\_place.tcl**

####################################################################

# Innovus Foundation Flow Code Generator, Sun Nov 24 21:55:33 CST 2019

# Version : 17.10-p003\_1

####################################################################

if {[file exists FF/vars.tcl]} {

source FF/vars.tcl

}

foreach file $vars(config\_files) {

source $file

}

source FF/procs.tcl

ff\_procs::system\_info

setDistributeHost -local

setMultiCpuUsage -localCpu 8 -remoteHost 0 -cpuPerRemoteHost 1

#-------------------------------------------------------------

set vars(step) partition\_place

set vars(partition\_place,start\_time) [clock seconds]

um::push\_snapshot\_stack

#-------------------------------------------------------------

######################################################################

# Variables affecting this step:

#---------------------------------------------------------------------

# - vars(report\_run\_time)

# - vars(fp\_file)

# - vars(add\_tracks)

# - vars(max\_route\_layer)

# - vars(process)

######################################################################

# Additional variables for this step:

#---------------------------------------------------------------------

# - vars(ilm\_list) "<list of ILMs>"

# - vars(<ilm>,ilm\_dir) "<path to ILM directory>"

# - vars(<ilm>,lef\_file) "<LEF file for block associated with ILM>"

# - vars(<ilm>,setup\_lib) "<LIB file for block associated with ILM>"

# - vars(fp\_tcl\_file)

# - vars(fp\_tcl\_proc)

# - vars(oa\_fp)

# - vars(def\_files)

# - vars(ilm\_non\_sdc\_file)

# - vars(activity\_file)

# - vars(activity\_file\_format)

# - vars(scan\_def)

# - vars(spare\_cells)

# - vars(dont\_use\_list)

# - vars(dont\_use\_file)

# - vars(use\_list)

# - vars(power\_effort) "low or high"

# - vars(welltaps)

# - vars(pre\_endcap)

# - vars(post\_endcap)

######################################################################

set vars(partition\_place,start\_time) [clock seconds]

exec mkdir -p $env(VPATH)

puts "<FF> Plugin -> pre\_init\_tcl"

source FF/init.tcl

init\_design

um::enable\_metrics -on

um::push\_snapshot\_stack

puts "<FF> Plugin -> always\_source\_tcl"

loadFPlan dtmf\_recvr\_core.fp

add\_tracks

source /home/yongfu/proj/jizuo/aeda/eda/work/check\_track.tcl

snapFPlan -all

snapFPlanIO

set\_power\_analysis\_mode -analysis\_view dtmf\_view\_slow

source FF/timingderate.sdc

setMaxRouteLayer 6

setDesignMode -process 45

puts "<FF> Plugin -> post\_init\_tcl"

checkDesign -all

check\_timing

timeDesign -prePlace -prefix preplace -outDir ./RPT

if {[dbGet -e top.ptns.clones] != ""} {

alignPtnClone

}

setPlaceMode -place\_global\_cong\_effort auto \

-place\_global\_clock\_gate\_aware true \

-place\_global\_place\_io\_pins true

######################################################################

# Variables affecting this step:

#---------------------------------------------------------------------

# - vars(process)

# - vars(place\_io\_pins)

# - vars(in\_place\_opt)

# - vars(leakage\_power\_effort)

# - vars(dynamic\_power\_effort)

######################################################################

# Additional variables for this step:

#---------------------------------------------------------------------

# - vars(power\_effort) "low or high"

# - vars(enable\_ocv) "pre\_postcts"

# - vars(enable\_aocv) "true"

# - vars(enable\_socv) "true"

# - vars(enable\_ss) "pre\_place"

# - vars(congestion\_effort) "auto low medium high"

# - vars(clock\_gate\_aware) "true"

# - vars(size\_only\_file) "<file>"

# - vars(clock\_gate\_aware\_opt) "true"

# - vars(all\_end\_points) "true"

# - vars(fix\_fanout\_load) "<file>"

# - vars(useful\_skew) "true"

# - vars(skew\_buffers) "<list of buffers>"

######################################################################

# The active analysis views are controlled by the following variables:

#---------------------------------------------------------------------

# - vars(place,active\_setup\_views)

# - vars(place,active\_hold\_views)

#

######################################################################

# set\_analysis\_view -setup $vars(place,active\_setup\_views) -hold $vars(place,active\_hold\_views)

#

setDesignMode -process 45

setOptMode -leakagePowerEffort high -dynamicPowerEffort high

Puts "<FF> RUNNING PLACEMENT ..."

puts "<FF> Plugin -> pre\_place\_tcl"

placeDesign

puts "<FF> Plugin -> post\_place\_tcl"

saveDesign ./DBS/place.enc -compress

earlyGlobalRoute

insertPtnFeedthrough -routeBased \

-saveTopoFile topo.txt \

-netMapping netmapping.txt \

-doubleBuffer

setRouteMode -earlyGlobalRoutePartitionHonorFence .

earlyGlobalRoute

timeDesign -prects -prefix place -outDir ./RPT

#-------------------------------------------------------------

report\_power -view dtmf\_view\_slow -outfile ./RPT/partition\_place.power.rpt

um::pop\_snapshot\_stack

create\_snapshot -name partition\_place -categories design

report\_metric -file ./RPT/metrics.html -format html

saveDesign ./DBS/partition\_place.enc -compress

saveNetlist ./DBS/LEC/partition\_place.v.gz

if {[info exists env(VPATH)]} {exec /bin/touch $env(VPATH)/partition\_place}

ff\_procs::report\_time

puts "<FF> Plugin -> final\_always\_source\_tcl"

if {![info exists vars(single)]} {

exit 0

}

**Run\_assign\_pin.tcl**

####################################################################

# Innovus Foundation Flow Code Generator, Sun Nov 24 21:55:33 CST 2019

# Version : 17.10-p003\_1

####################################################################

if {[file exists FF/vars.tcl]} {

source FF/vars.tcl

}

foreach file $vars(config\_files) {

source $file

}

source FF/procs.tcl

ff\_procs::system\_info

setDistributeHost -local

setMultiCpuUsage -localCpu 8 -remoteHost 0 -cpuPerRemoteHost 1

if {$vars(restore\_design)} { restoreDesign ./DBS/partition\_place.enc.dat dtmf\_recvr\_core }

um::enable\_metrics -on

puts "<FF> Plugin -> always\_source\_tcl"

#-------------------------------------------------------------

set vars(step) assign\_pin

set vars(assign\_pin,start\_time) [clock seconds]

um::push\_snapshot\_stack

#-------------------------------------------------------------

create\_snapshot -name assign\_pin -categories design

puts "<FF> Plugin -> pre\_assign\_pin\_tcl"

snapPtnPinsToTracks GROUP1

snapPtnPinsToTracks GROUP2

snapPtnPinsToTracks GROUP3

snapPtnPinsToTracks GROUP4

snapPtnPinsToTracks GROUP5

snapPtnPinsToTracks GROUP6

assignPtnPin

puts "<FF> Plugin -> post\_assign\_pin\_tcl"

checkPinAssignment -outFile pincheck.rpt

reportUnalignedNets -rptFile ./RPT/unaligned.rpt

setRouteMode -earlyGlobalRoutePartitionHonorFence . \

-earlyGlobalRoutePartitionHonorPin .

earlyGlobalRoute

#-------------------------------------------------------------

report\_power -view dtmf\_view\_slow -outfile ./RPT/assign\_pin.power.rpt

um::pop\_snapshot\_stack

create\_snapshot -name assign\_pin -categories design

report\_metric -file ./RPT/metrics.html -format html

saveDesign ./DBS/assign\_pin.enc -compress

saveNetlist ./DBS/LEC/assign\_pin.v.gz

if {[info exists env(VPATH)]} {exec /bin/touch $env(VPATH)/assign\_pin}

ff\_procs::report\_time

puts "<FF> Plugin -> final\_always\_source\_tcl"

if {![info exists vars(single)]} {

exit 0

}

**Run\_Partition.tcl**

####################################################################

# Innovus Foundation Flow Code Generator, Sun Nov 24 21:55:34 CST 2019

# Version : 17.10-p003\_1

####################################################################

if {[file exists FF/vars.tcl]} {

source FF/vars.tcl

}

foreach file $vars(config\_files) {

source $file

}

source FF/procs.tcl

ff\_procs::system\_info

setDistributeHost -local

setMultiCpuUsage -localCpu 4 -remoteHost 0 -cpuPerRemoteHost 1

if {$vars(restore\_design)} { restoreDesign ./DBS/assign\_pin.enc.dat dtmf\_recvr\_core }

um::enable\_metrics -on

puts "<FF> Plugin -> always\_source\_tcl"

#-------------------------------------------------------------

set vars(step) partition

set vars(partition,start\_time) [clock seconds]

um::push\_snapshot\_stack

#-------------------------------------------------------------

set\_global timing\_support\_hierarchical\_pin\_constraints true

set tbgPrintExceptionInfoInJustify 1

setBudgetingMode -writeLatencyPerClock true

if {[dbGet -e top.ptns.clones] != ""} {

setBudgetingMode -mergeClones true

}

deriveTimingBudget -justify

puts "<FF> Plugin -> pre\_partition\_tcl"

partition -buildScan

savePartition -dir PARTITION -def -scanDef

puts "<FF> Plugin -> post\_partition\_tcl"

#-------------------------------------------------------------

um::pop\_snapshot\_stack

create\_snapshot -name partition -categories design

report\_metric -file ./RPT/metrics.html -format html

if {[info exists env(VPATH)]} {exec /bin/touch $env(VPATH)/partition}

ff\_procs::report\_time

puts "<FF> Plugin -> final\_always\_source\_tcl"

if {![info exists vars(single)]} {

exit 0

}

**Blocks:**

**Run\_all.tcl**

#!/usr/bin/env tclsh

# -\*-TCL-\*-

#

###############################################################################

# CADENCE COPYRIGHT NOTICE

# ?2008-2013 Cadence Design Systems, Inc. All rights reserved.

#------------------------------------------------------------------------------

#

# This Foundation Flow is provided as an example of how to perform specialized

# tasks.

#

# This work may not be copied, re-published, uploaded, or distributed in any way,

# in any medium, whether in whole or in part, without prior written permission

# from Cadence. Notwithstanding any restrictions herein, subject to compliance

# with the terms and conditions of the Cadence software license agreement under

# which this material was provided, this material may be copied and internally

# distributed solely for internal purposes for use with Cadence tools.

#

# This work is Cadence intellectual property and may under no circumstances be

# given to third parties, neither in original nor in modified versions, without

# explicit written permission from Cadence. The information contained herein is

# the proprietary and confidential information of Cadence or its licensors, and

# is supplied subject to, and may be used only by Cadence's current customers

# in accordance with, a previously executed license agreement between Cadence

# and its customer.

#

#------------------------------------------------------------------------------

# THIS MATERIAL IS PROVIDED BY CADENCE "AS IS" AND ANY EXPRESS OR IMPLIED

# WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF

# MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.

# IN NO EVENT SHALL CADENCE BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL

# OR CONSEQUENTIAL DAMAGES HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY,

# WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR

# OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS MATERIAL, EVEN IF

# ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

###############################################################################

source FF/vars.tcl

if {![info exists env(VPATH)]} {

set env(VPATH) "make"

}

if {[info exists env(FF\_START)]} {

if {[file exists $env(VPATH)/$env(FF\_START)]} {

exec rm $env(VPATH)/$env(FF\_START)

}

set start\_step $env(FF\_START)

} else {

set start\_step init

}

if {[info exists env(FF\_STOP)]} {

set stop\_step $env(FF\_STOP)

} else {

set stop\_step signoff

}

#

# Start with all possible steps in the flow that we can execute. Then find

# the subset of steps that we generated a Tcl control file for

#

set vars(single) true

set found\_steps [list]

foreach step $vars(bsteps) {

set run\_file($step) "$vars(script\_dir)/INNOVUS/run\_$step.tcl"

if {[file exists $run\_file($step)]} {

lappend found\_steps $step

}

}

#set steps $found\_steps

set time 0

set index 0

foreach step $found\_steps {

if {[file exists $env(VPATH)/$step]} {

if {[file mtime $env(VPATH)/$step] > $time} {

if {[lindex $found\_steps $index] == "signoff"} {

puts "<FF> STEP signoff COMPLETE ... NOTHING TO DO"

exit

} else {

set start\_step [lindex $found\_steps [expr $index + 1]]

}

}

} else {

set start\_step $step

break

}

incr index

}

puts "<FF> -----------------------------------------------------"

if {$start\_step == $stop\_step} {

puts "<FF> RUNNING STEP $start\_step ..."

} else {

puts "<FF> RUNNING STEP $start\_step through $stop\_step ..."

}

puts "<FF> -----------------------------------------------------"

sleep 5

#

# Emulate the functionality of the "make" program. If a file doesn't exist,

# or if it's predecessor step has executed more recently, run the current

# step and update the time/date stamps of the files we track progress with

#

set vars(stop\_step) $stop\_step

set last\_step ""

set vars(single) 1

if {$vars(enable\_qor\_check)} {

set check\_qor $vars(script\_dir)/check\_qor.tcl

}

exec mkdir -p make

foreach step $found\_steps {

set vars(step) $step

puts "<FF> STEP $vars(step) ..."

set this\_semaphore "$env(VPATH)/$step"

set last\_semaphore "$env(VPATH)/$last\_step"

set run\_step false

if {![file exists $this\_semaphore]} {

set run\_step true

} elseif {$last\_step != ""} {

file stat $this\_semaphore this\_stat

file stat $last\_semaphore last\_stat

if {$this\_stat(mtime) < $last\_stat(mtime)} {

set run\_step true

}

}

#

# Check the quality of results from the last step. If the results aren't

# as expected, don't proceed to the next step

#

if {$vars(enable\_qor\_check)} {

if {$step != "init"} {

set qor [catch {exec $check\_qor $last\_semaphore} msg]

if {$qor} {

puts "<FF> ERROR: QOR RESULT CHECK INDICATES A PROBLEM IN STEP $last\_step"

exit -111

}

}

}

set last\_step $step

if {!$run\_step} {

continue

}

#

# Source the control file that runs the step. If it had an error, exit

# immediately. Otherwise, update the time/date stamp of the corresponding

#

#

if {[catch {source $run\_file($step)} message]} {

puts $message

exit -111

}

#

# If there is a QOR check file for this step, execute it and place the

# result into the make file. Otherwise just touch the file

#

if {$vars(enable\_qor\_check)} {

if {[info exists vars($step,qor\_tcl]} {

set qor\_file $vars($step,qor\_tcl)

if {[file exists $qor\_file]} {

set qor [catch {exec $qor\_file} msg]

set make\_file [open $this\_semaphore w]

puts $make\_file "$qor"

close $make\_file

}

}

}

exec /bin/touch $this\_semaphore

if {$step == $stop\_step} {

exit

}

}

**Makefile**

VERSION=17.10-p003\_1

VPATH=make

TCLSH=/home/yongfu/anaconda3/bin/tclsh

GEN\_FLOW=SCRIPTS/gen\_flow.tcl

SETUP\_PATH=PARTITION/GROUP1

TOOL=innovus

ARGS=-64 -nowin

FSTEPS=init place cts postcts\_hold route postroute signoff

SCRIPTS=FF

LOG=LOG

BROWSER=netscape

STEPS = version setup init place cts postcts\_hold route postroute signoff do\_cleanup

FF\_START = init

FF\_STOP = signoff

all: version setup init place cts postcts\_hold route postroute signoff do\_cleanup

version:

@echo "# Foundation Flows Version $(VERSION)"

help:

@echo "==================================================="

@echo " $(VERSION) Foundation Flows"

@echo "==================================================="

@echo " Makefile Targets"

@echo "==================================================="

@echo " setup : Setup Run Directory"

@echo init : Design Import / Initialization

@echo place : Cell Placement

@echo cts : Clock Tree Synthesis

@echo postcts\_hold : PostCTS Hold Fixing

@echo route : Global/Detail Route

@echo postroute : PostRoute Optimization

@echo signoff : Signoff Timing / Verify

@echo "---------------------------------------------------"

@echo " all : All design steps"

@echo " simple : Single script (all steps in a single session) - no stop/start"

@echo " single : Single script (all steps in a single session)"

@echo "==================================================="

@echo " Makefile Options"

@echo "==================================================="

@echo " VPATH : Make directory (default make)"

@echo " TOOL : INNOVUS executable (default innovus)"

@echo " ARGS : INNOVUS arguments (default -nowin -64)"

@echo " SCRIPTS : Script directory (default FF)"

@echo " LOG : Logfile directory (default LOG)"

@echo "==================================================="

simple: setup

FF\_STOP=$(FF\_STOP); VPATH=$(VPATH); export VPATH; $(TOOL) -init $(SCRIPTS)/INNOVUS/run\_simple.tcl -log $(LOG)/simple.log $(ARGS)

single: setup

FF\_STOP=$(FF\_STOP); VPATH=$(VPATH); export VPATH; $(TOOL) -init $(SCRIPTS)/INNOVUS/run\_all.tcl -log $(LOG)/single.log $(ARGS)

setup:

@/bin/mkdir -p $(VPATH) $(LOG)

@/bin/touch $(VPATH)/$@

init: setup

@mkdir -p $(LOG);

VPATH=$(VPATH); export VPATH; $(TOOL) -init $(SCRIPTS)/INNOVUS/run\_init.tcl -log $(LOG)/init.log -overwrite $(ARGS)

place: init

@mkdir -p $(LOG);

VPATH=$(VPATH); export VPATH; $(TOOL) -init $(SCRIPTS)/INNOVUS/run\_place.tcl -log $(LOG)/place.log -overwrite $(ARGS)

cts: place

@mkdir -p $(LOG);

VPATH=$(VPATH); export VPATH; $(TOOL) -init $(SCRIPTS)/INNOVUS/run\_cts.tcl -log $(LOG)/cts.log -overwrite $(ARGS)

postcts\_hold: cts

@mkdir -p $(LOG);

VPATH=$(VPATH); export VPATH; $(TOOL) -init $(SCRIPTS)/INNOVUS/run\_postcts\_hold.tcl -log $(LOG)/postcts\_hold.log -overwrite $(ARGS)

route: postcts\_hold

@mkdir -p $(LOG);

VPATH=$(VPATH); export VPATH; $(TOOL) -init $(SCRIPTS)/INNOVUS/run\_route.tcl -log $(LOG)/route.log -overwrite $(ARGS)

postroute: route

@mkdir -p $(LOG);

VPATH=$(VPATH); export VPATH; $(TOOL) -init $(SCRIPTS)/INNOVUS/run\_postroute.tcl -log $(LOG)/postroute.log -overwrite $(ARGS)

signoff: postroute

@mkdir -p $(LOG);

VPATH=$(VPATH); export VPATH; $(TOOL) -init $(SCRIPTS)/INNOVUS/run\_signoff.tcl -log $(LOG)/signoff.log -overwrite $(ARGS)

debug\_%:

VPATH=$(VPATH); export STEP=$\* ; export VPATH; $(TOOL) -init $(SCRIPTS)/INNOVUS/run\_debug.tcl -log $(LOG)/$@.log -win $(ARGS:-nowin=)

lec\_%:

export STEP=$\* ; lec -64 -xl -Dofile $(SCRIPTS)/INNOVUS/run\_lec.tcl -NOGui -LOGfile $(LOG)/$@.log

help\_%:

/home/yongfu/anaconda3/bin/tclsh $(GEN\_FLOW) -H $\*

.PHONY: clean

clean:

/bin/mv \*.rpt ./RPT ;\

/bin/rm -fr extLogDir\* \_\_qrc.log \*cts\_trace \*.rpt.old \*delete\* placementReports\* \*.rguide \*\_mmmc \

\*\_constr.pt .constr\* .FE\* .routing\* .timing\_file\* .tdrlog\*

do\_cleanup: signoff

$(MAKE) clean

/bin/touch $(VPATH)/$@

reset : version

/bin/rm -fr $(VPATH)/\* extLogDir\* \_\_qrc.log \*cts\_trace \*.rpt.old \*delete\* placementReports\* \*.rguide \*\_mmmc

@for file in $(STEPS) ; \

do \

if [ -r $(VPATH)/$$file ] ; then \

/bin/rm $(VPATH)/$$file ; \

fi \

done

block\_%: setup

@if [ "x$\*" = "xsingle" ] ; then \

ff\_stop=$(FF\_STOP); \

target="$@ (from: $(FF\_START) to: $(FF\_STOP))"; \

else \

ff\_stop=$\* ; \

target=$@; \

fi; \

if [ -r $(VPATH)/.RUNNING ] ; then \

echo "INFO: A build seems to be running already... check $(VPATH)/.RUNNING file and remove that file if the process is dead" ; \

/bin/head -1 $(VPATH)/.RUNNING ; \

exit -1 ; \

else \

/bin/rm -f $(VPATH)/block\_$${ff\_stop}.DONE $(VPATH)/block\_$${ff\_stop}.FAILED $(VPATH)/block\_$${ff\_stop}.PASS ; \

(echo "# Started building $${target} at "`/bin/date`" on "`/bin/hostname`" PID: $$$$" ; $(MAKE) $(TARGET) ) &>$(VPATH)/.RUNNING ; \

if [ -r $(VPATH)/$${ff\_stop} ] ; then \

if [ -r $(VPATH)/.RUNNING ] ; then \

/bin/mv $(VPATH)/.RUNNING $(VPATH)/block\_$${ff\_stop}.PASS ; \

/bin/touch $(VPATH)/block\_$${ff\_stop} ; \

/bin/touch $(VPATH)/block\_$${ff\_stop}.DONE ; \

else \

echo "# Something did not work properly" > $(VPATH)/block\_$${ff\_stop}.FAILED ; \

/bin/touch $(VPATH)/block\_$${ff\_stop}.DONE ; \

exit -1; \

fi ; \

else \

if [ -r $(VPATH)/.RUNNING ] ; then \

/bin/mv $(VPATH)/.RUNNING $(VPATH)/block\_$${ff\_stop}.FAILED ; \

else \

echo "# Something did not work properly" > $(VPATH)/block\_$${ff\_stop}.FAILED ; \

fi ; \

/bin/touch $(VPATH)/block\_$${ff\_stop}.DONE ; \

exit -1 ; \

fi ; \

fi

**run\_init.tcl**

####################################################################

# Innovus Foundation Flow Code Generator, Sun Nov 24 21:55:33 CST 2019

# Version : 17.10-p003\_1

####################################################################

if {[file exists FF/vars.tcl]} {

source FF/vars.tcl

}

source FF/procs.tcl

ff\_procs::system\_info

setDistributeHost -local

setMultiCpuUsage -localCpu 8 -remoteHost 0 -cpuPerRemoteHost 1

######################################################################

# Variables affecting this step:

#---------------------------------------------------------------------

# - vars(report\_run\_time)

######################################################################

# Additional variables for this step:

#---------------------------------------------------------------------

# - vars(max\_route\_layer)

# - vars(cpf\_file)

# - vars(ieee1801\_file)

######################################################################

set vars(step) init

set vars(init,start\_time) [clock seconds]

exec mkdir -p $env(VPATH)

puts "<FF> Plugin -> pre\_init\_tcl"

#------------------------------------------------------------

# PARTITION (GROUP1) IMPLEMENTATION FLOW

#------------------------------------------------------------

restoreDesign . GROUP1

um::enable\_metrics -on

um::push\_snapshot\_stack

puts "<FF> Plugin -> always\_source\_tcl"

puts "<FF> Plugin -> post\_init\_tcl"

timeDesign -preplace -prefix preplace -outDir ./RPT

checkDesign -all

check\_timing

#-------------------------------------------------------------

report\_power -view dtmf\_view\_slow -outfile ./RPT/init.power.rpt

um::pop\_snapshot\_stack

create\_snapshot -name init -categories design

report\_metric -file ./RPT/metrics.html -format html

saveDesign ./DBS/init.enc -compress

saveNetlist ./DBS/LEC/init.v.gz

if {[info exists env(VPATH)]} {exec /bin/touch $env(VPATH)/init}

ff\_procs::report\_time

puts "<FF> Plugin -> final\_always\_source\_tcl"

if {![info exists vars(single)]} {

exit 0

}

**Run\_place.tcl**

####################################################################

# Innovus Foundation Flow Code Generator, Sun Nov 24 21:55:33 CST 2019

# Version : 17.10-p003\_1

####################################################################

if {[file exists FF/vars.tcl]} {

source FF/vars.tcl

}

source FF/procs.tcl

ff\_procs::system\_info

setDistributeHost -local

setMultiCpuUsage -localCpu 8 -remoteHost 0 -cpuPerRemoteHost 1

if {$vars(restore\_design)} { restoreDesign ./DBS/init.enc.dat GROUP1 }

um::enable\_metrics -on

puts "<FF> Plugin -> always\_source\_tcl"

#-------------------------------------------------------------

set vars(step) place

set vars(place,start\_time) [clock seconds]

um::push\_snapshot\_stack

#-------------------------------------------------------------

######################################################################

# Variables affecting this step:

#---------------------------------------------------------------------

# - vars(process)

# - vars(place\_io\_pins)

# - vars(in\_place\_opt)

# - vars(leakage\_power\_effort)

# - vars(dynamic\_power\_effort)

######################################################################

# Additional variables for this step:

#---------------------------------------------------------------------

# - vars(power\_effort) "low or high"

# - vars(enable\_ocv) "pre\_postcts"

# - vars(enable\_aocv) "true"

# - vars(enable\_socv) "true"

# - vars(enable\_ss) "pre\_place"

# - vars(congestion\_effort) "auto low medium high"

# - vars(clock\_gate\_aware) "true"

# - vars(size\_only\_file) "<file>"

# - vars(clock\_gate\_aware\_opt) "true"

# - vars(all\_end\_points) "true"

# - vars(fix\_fanout\_load) "<file>"

# - vars(useful\_skew) "true"

# - vars(skew\_buffers) "<list of buffers>"

######################################################################

# The active analysis views are controlled by the following variables:

#---------------------------------------------------------------------

# - vars(place,active\_setup\_views)

# - vars(place,active\_hold\_views)

#

######################################################################

# set\_analysis\_view -setup $vars(place,active\_setup\_views) -hold $vars(place,active\_hold\_views)

#

setDesignMode -process 45

setPlaceMode -place\_global\_place\_io\_pins false

setOptMode -leakagePowerEffort high -dynamicPowerEffort high

Puts "<FF> RUNNING PLACEMENT ..."

puts "<FF> Plugin -> pre\_place\_tcl"

place\_opt\_design -out\_dir ./RPT -prefix place

place\_opt\_design -out\_dir ./RPT -prefix place -incremental

puts "<FF> Plugin -> post\_place\_tcl"

#-------------------------------------------------------------

report\_power -view dtmf\_view\_slow -outfile ./RPT/place.power.rpt

um::pop\_snapshot\_stack

create\_snapshot -name place -categories design

report\_metric -file ./RPT/metrics.html -format html

saveDesign ./DBS/place.enc -compress

saveNetlist ./DBS/LEC/place.v.gz

if {[info exists env(VPATH)]} {exec /bin/touch $env(VPATH)/place}

ff\_procs::report\_time

puts "<FF> Plugin -> final\_always\_source\_tcl"

if {![info exists vars(single)]} {

exit 0

}

**Run\_cts.tcl**

####################################################################

# Innovus Foundation Flow Code Generator, Sun Nov 24 21:55:33 CST 2019

# Version : 17.10-p003\_1

####################################################################

if {[file exists FF/vars.tcl]} {

source FF/vars.tcl

}

source FF/procs.tcl

ff\_procs::system\_info

setDistributeHost -local

setMultiCpuUsage -localCpu 8 -remoteHost 0 -cpuPerRemoteHost 1

if {$vars(restore\_design)} { restoreDesign ./DBS/place.enc.dat GROUP1 }

um::enable\_metrics -on

puts "<FF> Plugin -> always\_source\_tcl"

#-------------------------------------------------------------

set vars(step) cts

set vars(cts,start\_time) [clock seconds]

um::push\_snapshot\_stack

#-------------------------------------------------------------

######################################################################

# Variables affecting this step:

#---------------------------------------------------------------------

# - vars(process)

# - vars(enable\_cppr)

# - vars(route\_clock\_nets)

# - vars(ccopt\_integration)

# - vars(update\_io\_latency)

# - vars(ccopt\_effort)

# - vars(multi\_cut\_effort)

# - vars(litho\_driven\_routing)

######################################################################

# Additional variables for this step:

#---------------------------------------------------------------------

# - vars(power\_effort) "low or high"

# - vars(enable\_ocv) "pre\_cts"

# - vars(enable\_aocv) "true"

# - vars(enable\_socv) "true"

# - vars(clk\_tree\_top\_layer)

# - vars(clk\_leaf\_top\_layer)

# - vars(clk\_tree\_bottom\_layer)

# - vars(clk\_leaf\_bottom\_layer)

# - vars(clk\_tree\_ndr)

# - vars(clk\_tree\_extra\_space)

# - vars(clk\_leaf\_ndr)

# - vars(clk\_leaf\_extra\_space)

# - vars(cts\_inverter\_cells)

# - vars(cts\_buffer\_cells)

# - vars(clock\_gate\_cells)

# - vars(cts\_use\_inverters)

# - vars(cts\_io\_opt)

# - vars(cts\_target\_skew)

# - vars(cts\_target\_slew)

# - vars(ccopt\_executable)

# - vars(clk\_tree\_shield\_net)

######################################################################

# The active analysis views are controlled by the following variables:

#---------------------------------------------------------------------

# - vars(cts,active\_setup\_views)

# - vars(cts,active\_hold\_views)

#

######################################################################

# set\_analysis\_view -setup $vars(cts,active\_setup\_views) -hold $vars(cts,active\_hold\_views)

#

setDesignMode -process 45

setAnalysisMode -cppr none

set\_ccopt\_mode -integration "native" -ccopt\_modify\_clock\_latency true

setNanoRouteMode -drouteUseMultiCutViaEffort high -routeWithLithoDriven FALSE -drouteOnGridOnly all -routeBottomRoutingLayer 2

Puts "<FF> RUNNING CLOCK TREE SYNTHESIS ..."

source FF/timingderate.sdc

puts "<FF> Plugin -> pre\_cts\_tcl"

create\_ccopt\_clock\_tree\_spec

ccopt\_design -outDir ./RPT -prefix cts

puts "<FF> Plugin -> post\_cts\_tcl"

#-------------------------------------------------------------

report\_power -view dtmf\_view\_slow -outfile ./RPT/cts.power.rpt

um::pop\_snapshot\_stack

create\_snapshot -name cts -categories design

report\_metric -file ./RPT/metrics.html -format html

saveDesign ./DBS/cts.enc -compress

saveNetlist ./DBS/LEC/cts.v.gz

if {[info exists env(VPATH)]} {exec /bin/touch $env(VPATH)/cts}

ff\_procs::report\_time

puts "<FF> Plugin -> final\_always\_source\_tcl"

if {![info exists vars(single)]} {

exit 0

}

**Run\_postcts.tcl**

####################################################################

# Innovus Foundation Flow Code Generator, Sun Nov 24 21:55:33 CST 2019

# Version : 17.10-p003\_1

####################################################################

if {[file exists FF/vars.tcl]} {

source FF/vars.tcl

}

source FF/procs.tcl

ff\_procs::system\_info

setDistributeHost -local

setMultiCpuUsage -localCpu 8 -remoteHost 0 -cpuPerRemoteHost 1

if {$vars(restore\_design)} { restoreDesign ./DBS/cts.enc.dat GROUP1 }

um::enable\_metrics -on

puts "<FF> Plugin -> always\_source\_tcl"

#-------------------------------------------------------------

set vars(step) postcts\_hold

set vars(postcts\_hold,start\_time) [clock seconds]

um::push\_snapshot\_stack

#-------------------------------------------------------------

######################################################################

# Variables affecting this step:

#---------------------------------------------------------------------

# - vars(process)

######################################################################

# Additional variables for this step:

#---------------------------------------------------------------------

# - vars(power\_effort) "low or high"

# - vars(delay\_cells)

# - vars(fix\_hold\_allow\_tns\_degradation)

# - vars(fix\_hold\_ignore\_ios)

######################################################################

# The active analysis views are controlled by the following variables:

#---------------------------------------------------------------------

# - vars(postcts\_hold,active\_setup\_views)

# - vars(postcts\_hold,active\_hold\_views)

#

######################################################################

# set\_analysis\_view -setup $vars(postcts\_hold,active\_setup\_views) -hold $vars(postcts\_hold,active\_hold\_views)

#

setDesignMode -process 45

setNanoRouteMode -drouteUseMultiCutViaEffort high -routeWithLithoDriven FALSE -drouteOnGridOnly all -routeBottomRoutingLayer 1

Puts "<FF> RUNNING POST-CTS HOLD FIXING ..."

puts "<FF> Plugin -> pre\_postcts\_hold\_tcl"

optDesign -postCTS -hold -outDir ./RPT -prefix postcts\_hold

puts "<FF> Plugin -> post\_postcts\_hold\_tcl"

#-------------------------------------------------------------

report\_power -view dtmf\_view\_slow -outfile ./RPT/postcts\_hold.power.rpt

um::pop\_snapshot\_stack

create\_snapshot -name postcts\_hold -categories design

report\_metric -file ./RPT/metrics.html -format html

saveDesign ./DBS/postcts\_hold.enc -compress

saveNetlist ./DBS/LEC/postcts\_hold.v.gz

if {[info exists env(VPATH)]} {exec /bin/touch $env(VPATH)/postcts\_hold}

ff\_procs::report\_time

puts "<FF> Plugin -> final\_always\_source\_tcl"

if {![info exists vars(single)]} {

exit 0

}

**Run\_route.tcl**

####################################################################

# Innovus Foundation Flow Code Generator, Sun Nov 24 21:55:33 CST 2019

# Version : 17.10-p003\_1

####################################################################

if {[file exists FF/vars.tcl]} {

source FF/vars.tcl

}

source FF/procs.tcl

ff\_procs::system\_info

setDistributeHost -local

setMultiCpuUsage -localCpu 16 -remoteHost 0 -cpuPerRemoteHost 1

if {$vars(restore\_design)} { restoreDesign ./DBS/postcts\_hold.enc.dat GROUP1 }

um::enable\_metrics -on

puts "<FF> Plugin -> always\_source\_tcl"

#-------------------------------------------------------------

set vars(step) route

set vars(route,start\_time) [clock seconds]

um::push\_snapshot\_stack

#-------------------------------------------------------------

######################################################################

# Variables affecting this step:

#---------------------------------------------------------------------

# - vars(enable\_cppr)

# - vars(track\_opt)

# - vars(enable\_si\_aware)

# - vars(multi\_cut\_effort)

# - vars(litho\_driven\_routing)

######################################################################

# Additional variables for this step:

#---------------------------------------------------------------------

# - vars(antenna\_diode)

######################################################################

# The active analysis views are controlled by the following variables:

#---------------------------------------------------------------------

# - vars(route,active\_setup\_views)

# - vars(route,active\_hold\_views)

#

######################################################################

# set\_analysis\_view -setup $vars(route,active\_setup\_views) -hold $vars(route,active\_hold\_views)

#

setAnalysisMode -cppr both

setDelayCalMode -siAware true -engine aae

setNanoRouteMode -drouteUseMultiCutViaEffort high -routeWithLithoDriven FALSE -drouteOnGridOnly all -routeBottomRoutingLayer 1 -droutePostRouteSpreadWire false

setNanoRouteMode -drouteOnGridOnly via[1:2]

Puts "<FF> RUNNING GLOBAL/DETAIL ROUTING ..."

puts "<FF> Plugin -> pre\_route\_tcl"

setAnalysisMode -analysisType onChipVariation

#setNanoRouteMode -droutePostRouteSpreadWire false

generateVias

routeDesign

setExtractRCMode -engine postRoute -effortLevel high

puts "<FF> Plugin -> post\_route\_tcl"

#-------------------------------------------------------------

report\_power -view dtmf\_view\_slow -outfile ./RPT/route.power.rpt

um::pop\_snapshot\_stack

create\_snapshot -name route -categories design

report\_metric -file ./RPT/metrics.html -format html

saveDesign ./DBS/route.enc -compress

saveNetlist ./DBS/LEC/route.v.gz

if {[info exists env(VPATH)]} {exec /bin/touch $env(VPATH)/route}

ff\_procs::report\_time

puts "<FF> Plugin -> final\_always\_source\_tcl"

if {![info exists vars(single)]} {

exit 0

}

**Run\_postroute.tcl**

####################################################################

# Innovus Foundation Flow Code Generator, Sun Nov 24 21:55:33 CST 2019

# Version : 17.10-p003\_1

####################################################################

if {[file exists FF/vars.tcl]} {

source FF/vars.tcl

}

source FF/procs.tcl

ff\_procs::system\_info

setDistributeHost -local

setMultiCpuUsage -localCpu 16 -remoteHost 0 -cpuPerRemoteHost 1

if {$vars(restore\_design)} { restoreDesign ./DBS/route.enc.dat GROUP1 }

um::enable\_metrics -on

puts "<FF> Plugin -> always\_source\_tcl"

#-------------------------------------------------------------

set vars(step) postroute

set vars(postroute,start\_time) [clock seconds]

um::push\_snapshot\_stack

#-------------------------------------------------------------

######################################################################

# Variables affecting this step:

#---------------------------------------------------------------------

# - vars(process)

# - vars(postroute\_extraction\_effort)

######################################################################

# Additional variables for this step:

#---------------------------------------------------------------------

# - vars(power\_effort) "low or high"

# - vars(total\_c\_thresh)

# - vars(relative\_c\_thresh)

# - vars(coupling\_c\_thresh)

# - vars(qrc\_layer\_map)

# - vars(delay\_cells)

# - vars(fix\_hold\_allow\_tns\_degradation)

# - vars(fix\_hold\_ignore\_ios)

######################################################################

# The active analysis views are controlled by the following variables:

#---------------------------------------------------------------------

# - vars(postroute\_hold,active\_setup\_views)

# - vars(postroute\_hold,active\_hold\_views)

#

######################################################################

# set\_analysis\_view -setup $vars(postroute\_hold,active\_setup\_views) -hold $vars(postroute\_hold,active\_hold\_views)

#

setDesignMode -process 45

setExtractRCMode -engine postRoute -effortLevel high

######################################################################

# Variables affecting this step:

#---------------------------------------------------------------------

# - vars(process)

# - vars(process)

# - vars(postroute\_extraction\_effort)

# - vars(enable\_ocv)

# - vars(enable\_cppr)

# - vars(enable\_si\_aware)

######################################################################

# Additional variables for this step:

#---------------------------------------------------------------------

# - vars(power\_effort) "low or high"

# - vars(total\_c\_thresh)

# - vars(relative\_c\_thresh)

# - vars(relative\_c\_thresh)

# - vars(qrc\_layer\_map)

# - vars(enable\_aocv)

# - vars(enable\_socv)

######################################################################

# The active analysis views are controlled by the following variables:

#---------------------------------------------------------------------

# - vars(postroute,active\_setup\_views)

# - vars(postroute,active\_hold\_views)

#

######################################################################

# set\_analysis\_view -setup $vars(postroute,active\_setup\_views) -hold $vars(postroute,active\_hold\_views)

#

setDesignMode -process 45

setExtractRCMode -engine postRoute -effortLevel high

setAnalysisMode -analysisType onChipVariation -cppr none

setDelayCalMode -siAware true -engine aae

setNanoRouteMode -drouteOnGridOnly all -routeBottomRoutingLayer 1

setNanoRouteMode -drouteOnGridOnly via[1:2]

Puts "<FF> RUNNING POST-ROUTE OPTIMIZATION ..."

puts "<FF> Plugin -> pre\_postroute\_tcl"

puts "<FF> Plugin -> pre\_postroute\_hold\_tcl"

optDesign -postRoute -outDir ./RPT -prefix postroute -setup -hold -drv

optDesign -postRoute -outDir ./RPT -prefix postroute -hold -incr

optDesign -postRoute -outDir ./RPT -prefix postroute -hold -incr

optDesign -postRoute -outDir ./RPT -prefix postroute -hold -incr

optDesign -postRoute -outDir ./RPT -prefix postroute -hold -incr

optDesign -postRoute -outDir ./RPT -prefix postroute -hold -incr

optDesign -postRoute -outDir ./RPT -prefix postroute -hold -incr

optDesign -postRoute -outDir ./RPT -prefix postroute -hold -incr

#optDesign -postRoute -outDir ./RPT -prefix postroute -drv

puts "<FF> Plugin -> post\_postroute\_tcl"

#-------------------------------------------------------------

report\_power -view dtmf\_view\_slow -outfile ./RPT/postroute.power.rpt

um::pop\_snapshot\_stack

create\_snapshot -name postroute -categories design

report\_metric -file ./RPT/metrics.html -format html

saveDesign ./DBS/postroute.enc -compress

saveNetlist ./DBS/LEC/postroute.v.gz

if {[info exists env(VPATH)]} {exec /bin/touch $env(VPATH)/postroute}

ff\_procs::report\_time

puts "<FF> Plugin -> final\_always\_source\_tcl"

if {![info exists vars(single)]} {

exit 0

}

**Run\_signoff.tcl**

####################################################################

# Innovus Foundation Flow Code Generator, Sun Nov 24 21:55:33 CST 2019

# Version : 17.10-p003\_1

####################################################################

if {[file exists FF/vars.tcl]} {

source FF/vars.tcl

}

source FF/procs.tcl

ff\_procs::system\_info

setDistributeHost -local

setMultiCpuUsage -localCpu 4 -remoteHost 0 -cpuPerRemoteHost 1

if {$vars(restore\_design)} { restoreDesign ./DBS/postroute.enc.dat GROUP1 }

um::enable\_metrics -on

puts "<FF> Plugin -> always\_source\_tcl"

#-------------------------------------------------------------

set vars(step) signoff

set vars(signoff,start\_time) [clock seconds]

um::push\_snapshot\_stack

#-------------------------------------------------------------

######################################################################

# Variables affecting this step:

#---------------------------------------------------------------------

# - vars(process)

# - vars(signoff\_extraction\_effort)

# - vars(si\_analysis\_type)

# - vars(enable\_ocv)

# - vars(enable\_cppr)

######################################################################

# Additional variables for this step:

#---------------------------------------------------------------------

# - vars(total\_c\_thresh)

# - vars(relative\_c\_thresh)

# - vars(coupling\_c\_thresh)

# - vars(qrc\_layer\_map)

# - vars(delta\_delay\_threshold)

# - vars(celtic\_settings)

# - vars(enable\_aocv)

# - vars(enable\_socv)

# - vars(enable\_ss) "pre\_signoff"

######################################################################

# The active analysis views are controlled by the following variables:

#---------------------------------------------------------------------

# - vars(signoff,active\_setup\_views)

# - vars(signoff,active\_hold\_views)

#

######################################################################

# set\_analysis\_view -setup $vars(signoff,active\_setup\_views) -hold $vars(signoff,active\_hold\_views)

#

setDesignMode -process 45

setExtractRCMode -coupled true -effortLevel high

setAnalysisMode -analysisType onChipVariation -cppr none

set vars(active\_rc\_corners) [list]

foreach view [concat [all\_setup\_analysis\_views] [all\_hold\_analysis\_views]] {

set corner [get\_delay\_corner [get\_analysis\_view $view -delay\_corner] \

-rc\_corner]

if {[lsearch $vars(active\_rc\_corners) $corner] == -1 } {

lappend vars(active\_rc\_corners) $corner

}

}

Puts "<FF> ACTIVE RC CORNER LIST: $vars(active\_rc\_corners)"

set empty\_corners [list]

foreach corner $vars(active\_rc\_corners) {

if {![file exists [get\_rc\_corner $corner -qx\_tech\_file]]} {

lappend empty\_corners $corner

}

}

if {[llength $empty\_corners] == 0} {

setExtractRCMode -engine postRoute -effortLevel high -coupled true

} else {

Puts "<FF> CAN'T RUN SIGNOFF EXTRACTION BECAUSE qx\_tech\_file IS NOT DEFINED FOR these corners: $empty\_corners"

setExtractRCMode -engine postRoute -effortLevel low -coupled true

}

puts "<FF> Plugin -> pre\_signoff\_tcl"

Puts "<FF> RUNNING FINAL SIGNOFF ..."

extractRC

foreach corner $vars(active\_rc\_corners) {

rcOut -rc\_corner $corner -spef $corner.spef.gz

}

timeDesign -prefix signoff -signoff -reportOnly -outDir ./RPT

timeDesign -prefix signoff -signoff -reportOnly -hold -outDir ./RPT

createInterfaceLogic -dir GROUP1.ilm

summaryReport -outDir ./RPT

verifyConnectivity -noAntenna

verify\_drc

verifyMetalDensity

verifyProcessAntenna -leffile GROUP1.antenna.lef

puts "<FF> Plugin -> post\_signoff\_tcl"

#-------------------------------------------------------------

report\_power -view dtmf\_view\_slow -outfile ./RPT/signoff.power.rpt

um::pop\_snapshot\_stack

create\_snapshot -name signoff -categories design

report\_metric -file ./RPT/metrics.html -format html

saveDesign ./DBS/signoff.enc -compress

saveNetlist ./DBS/LEC/signoff.v.gz

if {[info exists env(VPATH)]} {exec /bin/touch $env(VPATH)/signoff}

ff\_procs::report\_time

puts "<FF> Plugin -> final\_always\_source\_tcl"

if {![info exists vars(single)]} {

exit 0

}

**Run\_lec.tcl**

//

// Innovus Foundation Flow Code Generator, Sun Nov 24 21:55:33 CST 2019

//

tclmode

if {[file exists PARTITION/GROUP1/FF/vars.tcl]} {

source PARTITION/GROUP1/FF/vars.tcl

}

source FF/procs.tcl

vpxmode

set dofile abort exit

set undefined cell black\_box -noascend -both

tclmode

// setup variables

global vars

proc Puts {args} {

puts $args

}

if {[info exists env(STEP)] && [file exists ./DBS/LEC/$env(STEP).v.gz]} {

set revisedNetlist ./DBS/LEC/$env(STEP).v.gz

} else {

Puts "<FF> Failed to find netlist DBS/LEC/$env(STEP).v.gz"

}

vpx set compare option -threads 1

// read liberty cell definitions

vpx read library -statetable \

-both \

-liberty [list /home/yongfu/proj/jizuo/aeda/eda/LIB/gsclib045\_v3.5/timing/slow.lib \

/home/yongfu/proj/jizuo/aeda/eda/LIB/macro\_libs/CDK\_R512x16.lib \

/home/yongfu/proj/jizuo/aeda/eda/LIB/macro\_libs/CDK\_S128x16.lib \

/home/yongfu/proj/jizuo/aeda/eda/LIB/macro\_libs/CDK\_S256x16.lib \

/home/yongfu/proj/jizuo/aeda/eda/LIB/macro\_libs/pllclk\_slow.lib]

// read reference netlist

vpx read design -verilog \

-sensitive \

-golden /home/yongfu/proj/jizuo/aeda/eda/work/dtmf\_recvr\_core\_1123new.v

// read post-implementation netlist

vpx read design -verilog -sensitive -revised $revisedNetlist

//set top level

vpx set root module GROUP1 -both

puts "<FF> Plugin -> pre\_lec\_tcl"

vpxmode

report design data

report black box

set mapping method -name first

set flatten model -seq\_constant -seq\_constant\_x\_to 0

set flatten model -nodff\_to\_dlat\_zero -nodff\_to\_dlat\_feedback

set flatten model -gated\_clock

set system mode lec

add compare point -all

compare -gate\_to\_gate

usage

// vpx report compare datareport compare data -class nonequivalent \

-class abort \

-class notcompared

report verification -verbose

report statistics

tclmode

set points\_count [get\_compare\_points -count]

set diff\_count [get\_compare\_points -diff -count]

set abort\_count [get\_compare\_points -abort -count]

set unknown\_count [get\_compare\_points -unknown -count]

if {$points\_count == 0} {

puts "---------------------------------"

puts "ERROR: No compare points detected"

puts "---------------------------------"

}

if {$diff\_count > 0} {

puts "------------------------------------"

puts "ERROR: Different Key Points detected"

puts "------------------------------------"

}

if {$abort\_count > 0} {

puts "-----------------------------"

puts "ERROR: Abort Points detected "

puts "-----------------------------"

}

if {$unknown\_count > 0} {

puts "----------------------------------"

puts "ERROR: Unknown Key Points detected"

puts "----------------------------------"

}

puts "No of compare points = $points\_count"

puts "No of diff points = $diff\_count"

puts "No of abort points = $abort\_count"

puts "No of unknown points = $unknown\_count"

exit 0

**run\_debug.tcl**

####################################################################

# Innovus Foundation Flow Code Generator, Sun Nov 24 21:55:33 CST 2019

# Version : 17.10-p003\_1

####################################################################

if {[file exists FF/vars.tcl]} {

source FF/vars.tcl

}

source FF/procs.tcl

ff\_procs::system\_info

setDistributeHost -local

setMultiCpuUsage -localCpu 4 -remoteHost 0 -cpuPerRemoteHost 1

#-------------------------------------------------------------

set vars(step) debug

set vars(debug,start\_time) [clock seconds]

um::push\_snapshot\_stack

#-------------------------------------------------------------

global env

if {[info exists env(STEP)]} {

if {$vars(restore\_design)} { restoreDesign ./DBS/$env(STEP).enc.dat GROUP1 }

um::enable\_metrics -on

puts "<FF> Plugin -> always\_source\_tcl"

}

win

#-------------------------------------------------------------

**signoff**

**run\_assemble.tcl**

####################################################################

# Innovus Foundation Flow Code Generator, Sun Nov 24 21:55:33 CST 2019

# Version : 17.10-p003\_1

####################################################################

if {[file exists FF/vars.tcl]} {

source FF/vars.tcl

}

foreach file $vars(config\_files) {

source $file

}

source FF/procs.tcl

ff\_procs::system\_info

setDistributeHost -local

setMultiCpuUsage -localCpu 16 -remoteHost 0 -cpuPerRemoteHost 1

#-------------------------------------------------------------

set vars(step) assemble

set vars(assemble,start\_time) [clock seconds]

um::push\_snapshot\_stack

#-------------------------------------------------------------

set vars(step) assemble

puts "<FF> Plugin -> pre\_assemble\_tcl"

assembleDesign -mmmcFile FF/view\_definition.tcl \

-fe \

-topDir PARTITION/dtmf\_recvr\_core/DBS/signoff.enc.dat \

-blockDir PARTITION/GROUP1/DBS/signoff.enc.dat\

-blockDir PARTITION/GROUP2/DBS/signoff.enc.dat\

-blockDir PARTITION/GROUP3/DBS/signoff.enc.dat\

-blockDir PARTITION/GROUP4/DBS/signoff.enc.dat\

-blockDir PARTITION/GROUP5/DBS/signoff.enc.dat\

-blockDir PARTITION/GROUP6/DBS/signoff.enc.dat

source FF/timingderate.sdc

um::enable\_metrics -on

um::push\_snapshot\_stack

puts "<FF> Plugin -> always\_source\_tcl"

puts "<FF> Plugin -> post\_assemble\_tcl"

Puts "<FF> UPDATING TIMING ..."

set op [open prop.sdc w]

puts $op "set\_propagated\_clock \[all\_clocks\]"

close $op

if {[lsearch [all\_constraint\_modes] dtmf\_constraint\_slow] != -1} {

set vars(dtmf\_constraint\_slow,post\_cts\_sdc\_list) [concat /home/yongfu/proj/jizuo/aeda/eda/Constraints/mmmc/dtmf\_recvr\_core\_gate\_slow.sdc prop.sdc]

}

if {[lsearch [all\_constraint\_modes] dtmf\_constraint\_typical] != -1} {

set vars(dtmf\_constraint\_typical,post\_cts\_sdc\_list) [concat /home/yongfu/proj/jizuo/aeda/eda/Constraints/mmmc/dtmf\_recvr\_core\_gate\_typical.sdc prop.sdc]

}

if {[lsearch [all\_constraint\_modes] dtmf\_constraint\_fast] != -1} {

set vars(dtmf\_constraint\_fast,post\_cts\_sdc\_list) [concat /home/yongfu/proj/jizuo/aeda/eda/Constraints/mmmc/dtmf\_recvr\_core\_gate\_fast.sdc prop.sdc]

}

set restore [get\_global timing\_defer\_mmmc\_object\_updates]

set\_global timing\_defer\_mmmc\_object\_updates true

foreach mode [all\_constraint\_modes] {

if {[info exists vars($mode,post\_cts\_sdc\_list)]} {

update\_constraint\_mode -name $mode \

-sdc\_files $vars($mode,post\_cts\_sdc\_list)

} else {

foreach view [all\_analysis\_views] {

set m [regsub \_$view $mode ""]

if {[info exists vars($m,post\_cts\_sdc\_list)]} {

update\_constraint\_mode -name $mode \

-sdc\_files $vars($m,post\_cts\_sdc\_list)

}

}

}

}

set\_analysis\_view -update\_timing

set\_global timing\_defer\_mmmc\_object\_updates $restore

######################################################################

# Variables affecting this step:

#---------------------------------------------------------------------

# - vars(process)

# - vars(signoff\_extraction\_effort)

# - vars(si\_analysis\_type)

# - vars(enable\_ocv)

# - vars(enable\_cppr)

######################################################################

# Additional variables for this step:

#---------------------------------------------------------------------

# - vars(total\_c\_thresh)

# - vars(relative\_c\_thresh)

# - vars(coupling\_c\_thresh)

# - vars(qrc\_layer\_map)

# - vars(delta\_delay\_threshold)

# - vars(celtic\_settings)

# - vars(enable\_aocv)

# - vars(enable\_socv)

# - vars(enable\_ss) "pre\_signoff"

######################################################################

# The active analysis views are controlled by the following variables:

#---------------------------------------------------------------------

# - vars(signoff,active\_setup\_views)

# - vars(signoff,active\_hold\_views)

#

######################################################################

# set\_analysis\_view -setup $vars(signoff,active\_setup\_views) -hold $vars(signoff,active\_hold\_views)

#

setDesignMode -process 45

setExtractRCMode -coupled true -effortLevel high

setAnalysisMode -analysisType onChipVariation -cppr none

set vars(active\_rc\_corners) [list]

foreach view [concat [all\_setup\_analysis\_views] [all\_hold\_analysis\_views]] {

set corner [get\_delay\_corner [get\_analysis\_view $view -delay\_corner] \

-rc\_corner]

if {[lsearch $vars(active\_rc\_corners) $corner] == -1 } {

lappend vars(active\_rc\_corners) $corner

}

}

Puts "<FF> ACTIVE RC CORNER LIST: $vars(active\_rc\_corners)"

set empty\_corners [list]

foreach corner $vars(active\_rc\_corners) {

if {![file exists [get\_rc\_corner $corner -qx\_tech\_file]]} {

lappend empty\_corners $corner

}

}

if {[llength $empty\_corners] == 0} {

setExtractRCMode -engine postRoute -effortLevel high -coupled true

} else {

Puts "<FF> CAN'T RUN SIGNOFF EXTRACTION BECAUSE qx\_tech\_file IS NOT DEFINED FOR these corners: $empty\_corners"

setExtractRCMode -engine postRoute -effortLevel low -coupled true

}

puts "<FF> Plugin -> pre\_signoff\_tcl"

Puts "<FF> RUNNING FINAL SIGNOFF ..."

extractRC

foreach corner $vars(active\_rc\_corners) {

rcOut -rc\_corner $corner -spef $corner.spef.gz

}

timeDesign -prefix signoff -signoff -reportOnly -outDir ./RPT

timeDesign -prefix signoff -signoff -reportOnly -hold -outDir ./RPT

summaryReport -outDir ./RPT

verifyConnectivity -noAntenna

verify\_drc

verifyMetalDensity

verifyProcessAntenna

puts "<FF> Plugin -> post\_signoff\_tcl"

#-------------------------------------------------------------

report\_power -view dtmf\_view\_slow -outfile ./RPT/assemble.power.rpt

um::pop\_snapshot\_stack

create\_snapshot -name assemble -categories design

report\_metric -file ./RPT/metrics.html -format html

saveDesign ./DBS/assemble.enc -compress

saveNetlist ./DBS/LEC/assemble.v.gz

if {[info exists env(VPATH)]} {exec /bin/touch $env(VPATH)/assemble}

ff\_procs::report\_time

puts "<FF> Plugin -> final\_always\_source\_tcl"

if {![info exists vars(single)]} {

exit 0

}

**run\_simple.tcl**

#####################################################################

# SINGLE SCRIPT FLOW

#####################################################################

source FF/vars.tcl

source FF/procs.tcl

ff\_procs::system\_info

setDistributeHost -local

setMultiCpuUsage -localCpu 4 -remoteHost 0 -cpuPerRemoteHost 1

#-------------------------------------------------------------

set vars(step) assemble

set vars(assemble,start\_time) [clock seconds]

um::push\_snapshot\_stack

#-------------------------------------------------------------

set vars(step) assemble

puts "<FF> Plugin -> pre\_assemble\_tcl"

assembleDesign -mmmcFile FF/view\_definition.tcl \

-fe \

-topDir PARTITION/dtmf\_recvr\_core/DBS/signoff.enc.dat \

-blockDir PARTITION/GROUP1/DBS/signoff.enc.dat\

-blockDir PARTITION/GROUP2/DBS/signoff.enc.dat\

-blockDir PARTITION/GROUP3/DBS/signoff.enc.dat\

-blockDir PARTITION/GROUP4/DBS/signoff.enc.dat\

-blockDir PARTITION/GROUP5/DBS/signoff.enc.dat\

-blockDir PARTITION/GROUP6/DBS/signoff.enc.dat

source FF/timingderate.sdc

um::enable\_metrics -on

um::push\_snapshot\_stack

puts "<FF> Plugin -> always\_source\_tcl"

puts "<FF> Plugin -> post\_assemble\_tcl"

Puts "<FF> UPDATING TIMING ..."

set op [open prop.sdc w]

puts $op "set\_propagated\_clock \[all\_clocks\]"

close $op

if {[lsearch [all\_constraint\_modes] dtmf\_constraint\_slow] != -1} {

set vars(dtmf\_constraint\_slow,post\_cts\_sdc\_list) [concat /home/yongfu/proj/jizuo/aeda/eda/Constraints/mmmc/dtmf\_recvr\_core\_gate\_slow.sdc prop.sdc]

}

if {[lsearch [all\_constraint\_modes] dtmf\_constraint\_typical] != -1} {

set vars(dtmf\_constraint\_typical,post\_cts\_sdc\_list) [concat /home/yongfu/proj/jizuo/aeda/eda/Constraints/mmmc/dtmf\_recvr\_core\_gate\_typical.sdc prop.sdc]

}

if {[lsearch [all\_constraint\_modes] dtmf\_constraint\_fast] != -1} {

set vars(dtmf\_constraint\_fast,post\_cts\_sdc\_list) [concat /home/yongfu/proj/jizuo/aeda/eda/Constraints/mmmc/dtmf\_recvr\_core\_gate\_fast.sdc prop.sdc]

}

set restore [get\_global timing\_defer\_mmmc\_object\_updates]

set\_global timing\_defer\_mmmc\_object\_updates true

foreach mode [all\_constraint\_modes] {

if {[info exists vars($mode,post\_cts\_sdc\_list)]} {

update\_constraint\_mode -name $mode \

-sdc\_files $vars($mode,post\_cts\_sdc\_list)

} else {

foreach view [all\_analysis\_views] {

set m [regsub \_$view $mode ""]

if {[info exists vars($m,post\_cts\_sdc\_list)]} {

update\_constraint\_mode -name $mode \

-sdc\_files $vars($m,post\_cts\_sdc\_list)

}

}

}

}

set\_analysis\_view -update\_timing

set\_global timing\_defer\_mmmc\_object\_updates $restore

######################################################################

# Variables affecting this step:

#---------------------------------------------------------------------

# - vars(process)

# - vars(signoff\_extraction\_effort)

# - vars(si\_analysis\_type)

# - vars(enable\_ocv)

# - vars(enable\_cppr)

######################################################################

# Additional variables for this step:

#---------------------------------------------------------------------

# - vars(total\_c\_thresh)

# - vars(relative\_c\_thresh)

# - vars(coupling\_c\_thresh)

# - vars(qrc\_layer\_map)

# - vars(delta\_delay\_threshold)

# - vars(celtic\_settings)

# - vars(enable\_aocv)

# - vars(enable\_socv)

# - vars(enable\_ss) "pre\_signoff"

######################################################################

# The active analysis views are controlled by the following variables:

#---------------------------------------------------------------------

# - vars(signoff,active\_setup\_views)

# - vars(signoff,active\_hold\_views)

#

######################################################################

# set\_analysis\_view -setup $vars(signoff,active\_setup\_views) -hold $vars(signoff,active\_hold\_views)

#

setDesignMode -process 45

setExtractRCMode -coupled true -effortLevel high

setAnalysisMode -analysisType onChipVariation -cppr none

set vars(active\_rc\_corners) [list]

foreach view [concat [all\_setup\_analysis\_views] [all\_hold\_analysis\_views]] {

set corner [get\_delay\_corner [get\_analysis\_view $view -delay\_corner] \

-rc\_corner]

if {[lsearch $vars(active\_rc\_corners) $corner] == -1 } {

lappend vars(active\_rc\_corners) $corner

}

}

Puts "<FF> ACTIVE RC CORNER LIST: $vars(active\_rc\_corners)"

set empty\_corners [list]

foreach corner $vars(active\_rc\_corners) {

if {![file exists [get\_rc\_corner $corner -qx\_tech\_file]]} {

lappend empty\_corners $corner

}

}

if {[llength $empty\_corners] == 0} {

setExtractRCMode -engine postRoute -effortLevel high -coupled true

} else {

Puts "<FF> CAN'T RUN SIGNOFF EXTRACTION BECAUSE qx\_tech\_file IS NOT DEFINED FOR these corners: $empty\_corners"

setExtractRCMode -engine postRoute -effortLevel low -coupled true

}

puts "<FF> Plugin -> pre\_signoff\_tcl"

Puts "<FF> RUNNING FINAL SIGNOFF ..."

extractRC

foreach corner $vars(active\_rc\_corners) {

rcOut -rc\_corner $corner -spef $corner.spef.gz

}

timeDesign -prefix signoff -signoff -reportOnly -outDir ./RPT

timeDesign -prefix signoff -signoff -reportOnly -hold -outDir ./RPT

summaryReport -outDir ./RPT

verifyConnectivity -noAntenna

verify\_drc

verifyMetalDensity

verifyProcessAntenna

puts "<FF> Plugin -> post\_signoff\_tcl"

#-------------------------------------------------------------

report\_power -view dtmf\_view\_slow -outfile ./RPT/assemble.power.rpt

um::pop\_snapshot\_stack

create\_snapshot -name assemble -categories design

report\_metric -file ./RPT/metrics.html -format html

saveDesign ./DBS/assemble.enc -compress

saveNetlist ./DBS/LEC/assemble.v.gz

if {[info exists env(VPATH)]} {exec /bin/touch $env(VPATH)/assemble}

ff\_procs::report\_time

puts "<FF> Plugin -> final\_always\_source\_tcl"

#-------------------------------------------------------------

set vars(step) partition\_place

set vars(partition\_place,start\_time) [clock seconds]

um::push\_snapshot\_stack

#-------------------------------------------------------------

######################################################################

# Variables affecting this step:

#---------------------------------------------------------------------

# - vars(report\_run\_time)

# - vars(fp\_file)

# - vars(add\_tracks)

# - vars(max\_route\_layer)

# - vars(process)

######################################################################

# Additional variables for this step:

#---------------------------------------------------------------------

# - vars(ilm\_list) "<list of ILMs>"

# - vars(<ilm>,ilm\_dir) "<path to ILM directory>"

# - vars(<ilm>,lef\_file) "<LEF file for block associated with ILM>"

# - vars(<ilm>,setup\_lib) "<LIB file for block associated with ILM>"

# - vars(fp\_tcl\_file)

# - vars(fp\_tcl\_proc)

# - vars(oa\_fp)

# - vars(def\_files)

# - vars(ilm\_non\_sdc\_file)

# - vars(activity\_file)

# - vars(activity\_file\_format)

# - vars(scan\_def)

# - vars(spare\_cells)

# - vars(dont\_use\_list)

# - vars(dont\_use\_file)

# - vars(use\_list)

# - vars(power\_effort) "low or high"

# - vars(welltaps)

# - vars(pre\_endcap)

# - vars(post\_endcap)

######################################################################

set vars(partition\_place,start\_time) [clock seconds]

exec mkdir -p $env(VPATH)

puts "<FF> Plugin -> pre\_init\_tcl"

source FF/init.tcl

init\_design

um::enable\_metrics -on

um::push\_snapshot\_stack

puts "<FF> Plugin -> always\_source\_tcl"

loadFPlan dtmf\_recvr\_core.fp

add\_tracks

set\_power\_analysis\_mode -analysis\_view dtmf\_view\_slow

source FF/timingderate.sdc

setMaxRouteLayer 6

setDesignMode -process 45

puts "<FF> Plugin -> post\_init\_tcl"

checkDesign -all

check\_timing

timeDesign -prePlace -prefix preplace -outDir ./RPT

if {[dbGet -e top.ptns.clones] != ""} {

alignPtnClone

}

setPlaceMode -place\_global\_cong\_effort auto \

-place\_global\_clock\_gate\_aware true \

-place\_global\_place\_io\_pins true

######################################################################

# Variables affecting this step:

#---------------------------------------------------------------------

# - vars(process)

# - vars(place\_io\_pins)

# - vars(in\_place\_opt)

# - vars(leakage\_power\_effort)

# - vars(dynamic\_power\_effort)

######################################################################

# Additional variables for this step:

#---------------------------------------------------------------------

# - vars(power\_effort) "low or high"

# - vars(enable\_ocv) "pre\_postcts"

# - vars(enable\_aocv) "true"

# - vars(enable\_socv) "true"

# - vars(enable\_ss) "pre\_place"

# - vars(congestion\_effort) "auto low medium high"

# - vars(clock\_gate\_aware) "true"

# - vars(size\_only\_file) "<file>"

# - vars(clock\_gate\_aware\_opt) "true"

# - vars(all\_end\_points) "true"

# - vars(fix\_fanout\_load) "<file>"

# - vars(useful\_skew) "true"

# - vars(skew\_buffers) "<list of buffers>"

######################################################################

# The active analysis views are controlled by the following variables:

#---------------------------------------------------------------------

# - vars(place,active\_setup\_views)

# - vars(place,active\_hold\_views)

#

######################################################################

# set\_analysis\_view -setup $vars(place,active\_setup\_views) -hold $vars(place,active\_hold\_views)

#

setDesignMode -process 45

setOptMode -leakagePowerEffort high -dynamicPowerEffort high

Puts "<FF> RUNNING PLACEMENT ..."

puts "<FF> Plugin -> pre\_place\_tcl"

placeDesign

puts "<FF> Plugin -> post\_place\_tcl"

saveDesign ./DBS/place.enc -compress

earlyGlobalRoute

insertPtnFeedthrough -routeBased \

-saveTopoFile topo.txt \

-netMapping netmapping.txt \

-doubleBuffer

setRouteMode -earlyGlobalRoutePartitionHonorFence .

earlyGlobalRoute

timeDesign -prects -prefix place -outDir ./RPT

#-------------------------------------------------------------

report\_power -view dtmf\_view\_slow -outfile ./RPT/partition\_place.power.rpt

um::pop\_snapshot\_stack

create\_snapshot -name partition\_place -categories design

report\_metric -file ./RPT/metrics.html -format html

saveDesign ./DBS/partition\_place.enc -compress

saveNetlist ./DBS/LEC/partition\_place.v.gz

if {[info exists env(VPATH)]} {exec /bin/touch $env(VPATH)/partition\_place}

ff\_procs::report\_time

puts "<FF> Plugin -> final\_always\_source\_tcl"

#-------------------------------------------------------------

set vars(step) assign\_pin

set vars(assign\_pin,start\_time) [clock seconds]

um::push\_snapshot\_stack

#-------------------------------------------------------------

puts "<FF> Plugin -> pre\_assign\_pin\_tcl"

assignPtnPin

puts "<FF> Plugin -> post\_assign\_pin\_tcl"

checkPinAssignment -outFile pincheck.rpt

reportUnalignedNets -rptFile ./RPT/unaligned.rpt

setRouteMode -earlyGlobalRoutePartitionHonorFence . \

-earlyGlobalRoutePartitionHonorPin .

earlyGlobalRoute

#-------------------------------------------------------------

report\_power -view dtmf\_view\_slow -outfile ./RPT/assign\_pin.power.rpt

um::pop\_snapshot\_stack

create\_snapshot -name assign\_pin -categories design

report\_metric -file ./RPT/metrics.html -format html

saveDesign ./DBS/assign\_pin.enc -compress

saveNetlist ./DBS/LEC/assign\_pin.v.gz

if {[info exists env(VPATH)]} {exec /bin/touch $env(VPATH)/assign\_pin}

ff\_procs::report\_time

puts "<FF> Plugin -> final\_always\_source\_tcl"

#-------------------------------------------------------------

set vars(step) partition

set vars(partition,start\_time) [clock seconds]

um::push\_snapshot\_stack

#-------------------------------------------------------------

set\_global timing\_support\_hierarchical\_pin\_constraints true

set tbgPrintExceptionInfoInJustify 1

setBudgetingMode -writeLatencyPerClock true

if {[dbGet -e top.ptns.clones] != ""} {

setBudgetingMode -mergeClones true

}

deriveTimingBudget -justify

puts "<FF> Plugin -> pre\_partition\_tcl"

partition -buildScan

savePartition -dir PARTITION -def -scanDef

puts "<FF> Plugin -> post\_partition\_tcl"

#-------------------------------------------------------------

um::pop\_snapshot\_stack

create\_snapshot -name partition -categories design

report\_metric -file ./RPT/metrics.html -format html

if {[info exists env(VPATH)]} {exec /bin/touch $env(VPATH)/partition}

ff\_procs::report\_time

puts "<FF> Plugin -> final\_always\_source\_tcl"

exit

**run\_lec.tcl**

//

// Innovus Foundation Flow Code Generator, Sun Nov 24 21:55:34 CST 2019

//

tclmode

if {[file exists FF/vars.tcl]} {

source FF/vars.tcl

}

source FF/procs.tcl

vpxmode

set dofile abort exit

set undefined cell black\_box -noascend -both

tclmode

// setup variables

global vars

proc Puts {args} {

puts $args

}

if {[info exists env(STEP)] && [file exists ./DBS/LEC/$env(STEP).v.gz]} {

set revisedNetlist ./DBS/LEC/$env(STEP).v.gz

} else {

Puts "<FF> Failed to find netlist DBS/LEC/$env(STEP).v.gz"

}

vpx set compare option -threads 1

// read liberty cell definitions

vpx read library -statetable \

-both \

-liberty [list /home/yongfu/proj/jizuo/aeda/eda/LIB/gsclib045\_v3.5/timing/slow.lib \

/home/yongfu/proj/jizuo/aeda/eda/LIB/macro\_libs/CDK\_R512x16.lib \

/home/yongfu/proj/jizuo/aeda/eda/LIB/macro\_libs/CDK\_S128x16.lib \

/home/yongfu/proj/jizuo/aeda/eda/LIB/macro\_libs/CDK\_S256x16.lib \

/home/yongfu/proj/jizuo/aeda/eda/LIB/macro\_libs/pllclk\_slow.lib]

// read reference netlist

vpx read design -verilog \

-sensitive \

-golden /home/yongfu/proj/jizuo/aeda/eda/work/dtmf\_recvr\_core\_1123new.v

// read post-implementation netlist

vpx read design -verilog -sensitive -revised $revisedNetlist

//set top level

vpx set root module dtmf\_recvr\_core -both

puts "<FF> Plugin -> pre\_lec\_tcl"

vpxmode

report design data

report black box

set mapping method -name first

set flatten model -seq\_constant -seq\_constant\_x\_to 0

set flatten model -nodff\_to\_dlat\_zero -nodff\_to\_dlat\_feedback

set flatten model -gated\_clock

set system mode lec

add compare point -all

compare -gate\_to\_gate

usage

// vpx report compare datareport compare data -class nonequivalent \

-class abort \

-class notcompared

report verification -verbose

report statistics

tclmode

set points\_count [get\_compare\_points -count]

set diff\_count [get\_compare\_points -diff -count]

set abort\_count [get\_compare\_points -abort -count]

set unknown\_count [get\_compare\_points -unknown -count]

if {$points\_count == 0} {

puts "---------------------------------"

puts "ERROR: No compare points detected"

puts "---------------------------------"

}

if {$diff\_count > 0} {

puts "------------------------------------"

puts "ERROR: Different Key Points detected"

puts "------------------------------------"

}

if {$abort\_count > 0} {

puts "-----------------------------"

puts "ERROR: Abort Points detected "

puts "-----------------------------"

}

if {$unknown\_count > 0} {

puts "----------------------------------"

puts "ERROR: Unknown Key Points detected"

puts "----------------------------------"

}

puts "No of compare points = $points\_count"

puts "No of diff points = $diff\_count"

puts "No of abort points = $abort\_count"

puts "No of unknown points = $unknown\_count"

exit 0

**run\_debug.tcl**

####################################################################

# Innovus Foundation Flow Code Generator, Sun Nov 24 21:55:34 CST 2019

# Version : 17.10-p003\_1

####################################################################

if {[file exists FF/vars.tcl]} {

source FF/vars.tcl

}

foreach file $vars(config\_files) {

source $file

}

source FF/procs.tcl

ff\_procs::system\_info

setDistributeHost -local

setMultiCpuUsage -localCpu 4 -remoteHost 0 -cpuPerRemoteHost 1

#-------------------------------------------------------------

set vars(step) debug

set vars(debug,start\_time) [clock seconds]

um::push\_snapshot\_stack

#-------------------------------------------------------------

global env

if {[info exists env(STEP)]} {

if {$vars(restore\_design)} { restoreDesign ./DBS/$env(STEP).enc.dat dtmf\_recvr\_core }

um::enable\_metrics -on

puts "<FF> Plugin -> always\_source\_tcl"

}

win

#-------------------------------------------------------------