

```
#####
# 2009-10-09 jmw:
# I have not been able to get some of the DC versions of 2008
# and 2009 to synthesize the Step07 FullDup design
# monolithically; sometimes, I had to do it unit-wise.
# Each new version of DC typically requires detailed tuning of
# the synthesis timing constraints.
# Here is a monolithic script which works with DC version
# C-2009.06-SP1 of July 2009.
#####
#
# The initial commands here are appropriate
# for use of the TSMC synthesis libraries:
#
set hdlin_translate_off_skip_text "true"
set verilayout_no_tri "true"
set default_schematic_options "-size infinite"
set write_name_nets_same_as_ports "true"
#
#####
# End TSMC commands.
#
# dc_shell Tcl startup script:
#
set designer "veriloguserN"
set company "SVTI"
#
# Some design environment variables:
#
set search_path ". [getenv SYNOPSYS]/../TSMCLibes/tcbn90ghpSYN"
#
# tc = Typical; bc = Best; wc = Worst:
set target_library tcbn90ghptc.db
set link_library tcbn90ghptc.db
#
set symbol_library tcbn90ghp.sdb
#
# -----
# Messages should be suppressed VERY CAREFULLY,
# and with foreknowledge of their meanings in
# previous DC invocations. These now are
# understood just to clutter the screen:
#
# The list suppressed will be presented below,
# just before compile.
#
set MessageList "VER-311"; # Parameter range specification is new.
set MessageList "$MessageList VER-314"; # Integers now are signed.
set MessageList "$MessageList TIM-112"; # Delay will be added for propagated clock.
set MessageList "$MessageList TIM-175"; # Breaking a timing path (user constraints).
set MessageList "$MessageList TIM-179"; # set_max_delay on pin no longer there.
set MessageList "$MessageList OPT-314"; # Disabling timing to break a loop.
set MessageList "$MessageList OPT-461"; # A dont_touch may be overridden: logic connected.
set MessageList "$MessageList OPT-1205"; # Register may not be optimal...library clock/enable phase.
set MessageList "$MessageList OPT-1206"; # Register is constant & will be removed.
set MessageList "$MessageList OPT-1307"; # Setting dont_use on a dont_touch'ed libe cell.
set MessageList "$MessageList LINT-63"; # Net ... has a single tri-state driver.
suppress_message $MessageList > /dev/null
set MessageList "
SUPPRESSED MESSAGES: $MessageList.
"
#
set MLimitList "LINT-2"; # Net ... driven by pin ... has no loads.
set MLimitList "$MLimitList LINT-28"; # Port ... is not connected to any nets.
foreach {M} "$MLimitList" { set_message_info -id $M -limit 20 };
set MLimitList "
LIMITED MESSAGES: $MLimitList.
"
#
# -----
#
define_design_lib FullDup -path ./FullDupSynth
#
```

```

analyze -work FullDup -format verilog FullDup.v
analyze -work FullDup -format verilog SerDes/SerDes.v
analyze -work FullDup -format verilog SerDes/Deserializer/DesDecoder/DesDecoder.v
analyze -work FullDup -format verilog SerDes/Deserializer/Deserializer.v
analyze -work FullDup -format verilog SerDes/Deserializer/SerialRx/SerialRx.v
analyze -work FullDup -format verilog SerDes/FIFO/DPMemlKx32.v
analyze -work FullDup -format verilog SerDes/FIFO/FIFOStateM.v
analyze -work FullDup -format verilog SerDes/FIFO/FIFOTop.v
analyze -work FullDup -format verilog SerDes/PLL/ClockComparator.v
analyze -work FullDup -format verilog SerDes/PLL/MultiCounter.v
analyze -work FullDup -format verilog SerDes/PLL/PLLTop.v
analyze -work FullDup -format verilog SerDes/PLL/VFO.v
analyze -work FullDup -format verilog SerDes/Serializer/SerEncoder/SerEncoder.v
analyze -work FullDup -format verilog SerDes/Serializer/Serializer.v
analyze -work FullDup -format verilog SerDes/Serializer/SerialTx/SerialTx.v
#
elaborate -work FullDup FullDup
#
# -----
#
set_operating_conditions NCCOM
set_wire_load_model -name "TSMC64K_Lowk_Conservative" [all_designs]
#
set_verilogout_higher_designs_first "true"
#
set_drive          3.0 [all_inputs]
set_load           10.0 [all_outputs]
set_max_fanout     15   [all_inputs]
set_max_fanout     20   [all_designs]
#
# module-specific constraints:
set_max_area 0
#
# Actual requirement = 1000:
set_ParPeriod 250
set_PLLRatio 32
set_SerPeriod [expr $ParPeriod/$PLLRatio]
#
#####
# Problem-node tuning parameters (delay divisors):
set_Speed1 25.0
set_Speed2 50.0
set_Speed3 100.0
#####
#
set_max_delay [expr $SerPeriod/8.0] -to [all_outputs]
set_max_delay [expr $SerPeriod/2.0] -from [all_inputs] -to [all_outputs]
#
set_Delta 1.0
set_RDelta [expr $Delta*8.0]
set_WDelta [expr $Delta*5.0]
set_PLLDelta $Delta
#
# -----
# Deserializer:
set_VFOClockIn1 [get_pins SerDes_U1/Des_U1/DesDec_Rx1/ParClk]
set_VFOClockOut1 [get_pins SerDes_U1/Des_U1/SerRx_Rx1/PLL_RxU1/VFO1/ClockOut]
set_ParOutClock1 [get_ports SerDes_U1/Des_U1/ParOutClk]
set_DecodeClockIn1 [get_pins SerDes_U1/Des_U1/DesDec_Rx1/SerClk]
set_VFOClockIn2 [get_pins SerDes_U2/Des_U1/DesDec_Rx1/ParClk]
set_VFOClockOut2 [get_pins SerDes_U2/Des_U1/SerRx_Rx1/PLL_RxU1/VFO1/ClockOut]
set_ParOutClock2 [get_ports SerDes_U2/Des_U1/ParOutClk]
set_DecodeClockIn2 [get_pins SerDes_U2/Des_U1/DesDec_Rx1/SerClk]
set_DesFIFOClock1 [get_pins SerDes_U1/Des_U1/FIFO_Rx1/Clk*]
set_DesFIFOClock2 [get_pins SerDes_U2/Des_U1/FIFO_Rx1/Clk*]
#
create_clock -period $ParPeriod $ParOutClock1
create_clock -period $ParPeriod $VFOClockIn1
create_generated_clock $VFOClockOut1 -source $VFOClockIn1 -multiply_by $PLLRatio
create_generated_clock $DecodeClockIn1 -source $VFOClockOut1 -multiply_by 1
create_clock -period $ParPeriod $ParOutClock2
create_clock -period $ParPeriod $VFOClockIn2
create_generated_clock $VFOClockOut2 -source $VFOClockIn2 -multiply_by $PLLRatio

```

```

create_generated_clock $DecodeClockIn2 -source $VFOClockOut2 -multiply_by 1
create_clock -period $ParPeriod $DesFIFOClock1
create_clock -period $ParPeriod $DesFIFOClock2
#
set_clock_uncertainty $RDelta [get_clock $ParOutClock1]
set_clock_uncertainty $WDelta [get_clock $VFOClockIn1]
set_clock_uncertainty $PLLDelta [get_clock $VFOClockOut1]
set_clock_uncertainty $PLLDelta [get_clock $DecodeClockIn1]
set_clock_uncertainty $RDelta [get_clock $ParOutClock2]
set_clock_uncertainty $WDelta [get_clock $VFOClockIn2]
set_clock_uncertainty $PLLDelta [get_clock $VFOClockOut2]
set_clock_uncertainty $PLLDelta [get_clock $DecodeClockIn2]
#
# Serializer:
set VFOClockInS1 [get_pins SerDes_U1/Ser_U1/SerEnc_Tx1/ParClk]
set VFOClockOutS1 [get_pins SerDes_U1/Ser_U1/SerTx_Tx1/PLL_TxU1/VF01/ClockOut]
set ParInClock1 [get_ports SerDes_U1/Ser_U1/ParInClk]
set VFOClockInS2 [get_pins SerDes_U2/Ser_U1/SerEnc_Tx1/ParClk]
set VFOClockOutS2 [get_pins SerDes_U2/Ser_U1/SerTx_Tx1/PLL_TxU1/VF01/ClockOut]
set ParInClock2 [get_ports SerDes_U2/Ser_U1/ParInClk]
set SerFIFOClock1 [get_pins SerDes_U1/Ser_U1/FIFO_Tx1/Clk*]
set SerFIFOClock2 [get_pins SerDes_U2/Ser_U1/FIFO_Tx1/Clk*]
#
create_clock -period $ParPeriod $ParInClock1
create_clock -period $ParPeriod $VFOClockInS1
create_generated_clock $VFOClockOutS1 -source $VFOClockInS1 -multiply_by $PLLRatio
create_clock -period $ParPeriod $ParInClock2
create_clock -period $ParPeriod $VFOClockInS2
create_generated_clock $VFOClockOutS2 -source $VFOClockInS2 -multiply_by $PLLRatio
create_clock -period $ParPeriod $SerFIFOClock1
create_clock -period $ParPeriod $SerFIFOClock2
#
set_clock_uncertainty $RDelta [get_clock $ParInClock1]
set_clock_uncertainty $WDelta [get_clock $VFOClockInS1]
set_clock_uncertainty $PLLDelta [get_clock $VFOClockOutS1]
set_clock_uncertainty $RDelta [get_clock $ParInClock2]
set_clock_uncertainty $WDelta [get_clock $VFOClockInS2]
set_clock_uncertainty $PLLDelta [get_clock $VFOClockOutS2]
#
# For flat-netlist readability:
set FIFO_Flags [get_nets {SerDes_U1/*F_Empty SerDes_U1/*F_Full \
                        SerDes_U2/*F_Empty SerDes_U2/*F_Full}];
set_dont_touch $FIFO_Flags;
set_dont_touch [get_nets {SerLine1 SerLine2}];
set_dont_touch [get_nets SerDes_U1/Des_U1/DesDec_Rx1/Count32*]
set_dont_touch [get_nets SerDes_U2/Des_U1/DesDec_Rx1/Count32*]
set_dont_touch [all_clocks];
#
# -----
# SerDes U1 Deserializer problem-signal constraints:
set_max_delay [expr $SerPeriod/$Speed1] -from [all_inputs] \
    -to SerDes_U1/Des_U1/FIFO_Rx1/FIFO_Mem1/ParityErr
set_max_delay [expr $SerPeriod/$Speed2] -from [all_inputs] \
    -to SerDes_U1/Des_U1/FIFO_Rx1/FIFO_Mem1/Storage*/*
set_max_delay [expr $SerPeriod/$Speed2] -from $DesFIFOClock1 \
    -to SerDes_U1/Des_U1/FIFO_Rx1/FIFO_Mem1/Storage*/*
set_max_delay [expr $SerPeriod/$Speed2] -from $DecodeClockIn1 \
    -to SerDes_U1/Des_U1/DesDec_Rx1/*FrameSR*/Q
set_max_delay [expr $SerPeriod/$Speed3] -from $DecodeClockIn1 \
    -to SerDes_U1/Des_U1/DesDec_Rx1/ParClk
set_max_delay [expr $SerPeriod/$Speed3] -from $DecodeClockIn1 \
    -to SerDes_U1/Des_U1/DesDec_Rx1/Count32*
#
# -----
# SerDes U2 Deserializer problem-signal constraints:
set_max_delay [expr $SerPeriod/$Speed1] -from [all_inputs] \
    -to SerDes_U2/Des_U1/FIFO_Rx1/FIFO_Mem1/ParityErr
set_max_delay [expr $SerPeriod/$Speed2] -from [all_inputs] \
    -to SerDes_U2/Des_U1/FIFO_Rx1/FIFO_Mem1/Storage*/*
set_max_delay [expr $SerPeriod/$Speed2] -from $DesFIFOClock2 \
    -to SerDes_U2/Des_U1/FIFO_Rx1/FIFO_Mem1/Storage*/*
set_max_delay [expr $SerPeriod/$Speed2] -from $DecodeClockIn2 \
    -to SerDes_U2/Des_U1/DesDec_Rx1/*FrameSR*/Q

```

```

set_max_delay [expr $$SerPeriod/$$Speed3] -from $DecodeClockIn2 \
  -to SerDes_U2/Des_U1/DesDec_Rx1/ParClk
set_max_delay [expr $$SerPeriod/$$Speed3] -from $DecodeClockIn2 \
  -to SerDes_U2/Des_U1/DesDec_Rx1/Count32*
#
# -----
# All Serializer problem-signal constraints:
set_max_delay [expr $$SerPeriod/$$Speed2] -from [all_inputs] \
  -to SerDes_U1/Ser_U1/FIFO_Tx1/FIFO_Mem1/ParityErr
set_max_delay [expr $$SerPeriod/$$Speed2] -from [all_inputs] \
  -to SerDes_U1/Ser_U1/FIFO_Tx1/FIFO_Mem1/Storage*/*
set_max_delay [expr $$SerPeriod/$$Speed2] -from $ParInClock1 \
  -to SerDes_U1/Ser_U1/SerEnc_Tx1/Sh_N*/Q
set_max_delay [expr $$SerPeriod/$$Speed2] -from $ParInClock1 \
  -to SerDes_U1/Ser_U1/SerEnc_Tx1/InBuf*/Q
set_max_delay [expr $$SerPeriod/$$Speed2] -from $SerFIFOClock1 \
  -to SerDes_U1/Ser_U1/FIFO_Tx1/FIFO_Mem1/Storage*/*
set_max_delay [expr $$SerPeriod/$$Speed2] -from [all_inputs] \
  -to SerDes_U2/Ser_U1/FIFO_Tx1/FIFO_Mem1/ParityErr
set_max_delay [expr $$SerPeriod/$$Speed2] -from [all_inputs] \
  -to SerDes_U2/Ser_U1/FIFO_Tx1/FIFO_Mem1/Storage*/*
set_max_delay [expr $$SerPeriod/$$Speed2] -from $ParInClock2 \
  -to SerDes_U2/Ser_U1/SerEnc_Tx1/Sh_N*/Q
set_max_delay [expr $$SerPeriod/$$Speed2] -from $ParInClock2 \
  -to SerDes_U2/Ser_U1/SerEnc_Tx1/InBuf*/Q
set_max_delay [expr $$SerPeriod/$$Speed2] -from $SerFIFOClock2 \
  -to SerDes_U2/Ser_U1/FIFO_Tx1/FIFO_Mem1/Storage*/*
#
# set_propagated_clock [all_clocks]
#
# Prevent unnecessary dangling outputs in hier netlist:
#
# To force fix of hold violations -- BUT, increases time by ~100x:
#set_fix_hold [all_clocks]
#
echo "$MessageList"
echo "$MLimitList"
#
compile
# No special improvement: compile -map_effort high
#
write -hierarchy -format verilog -output FullDupNetlist_Hier.v
write_sdf FullDupNetlist_Hier.sdf
#
report_area > /dev/null
report_area
report_area > FullDupNetlist_Hier.log
report_timing
report_timing >> FullDupNetlist_Hier.log
check_design
check_design >> FullDupNetlist_Hier.log
#
ungroup -all -flatten
compile -incremental_mapping
#
write -hierarchy -format verilog -output FullDupNetlist_Flat.v
write_sdf FullDupNetlist_Flat.sdf
#
report_area > /dev/null
report_area
report_area > FullDupNetlist_Flat.log
report_timing
report_timing >> FullDupNetlist_Flat.log
check_design
check_design >> FullDupNetlist_Flat.log
#
# Drop into interactive mode:
#
exit

```