# Design Guideline for Design Implementation

**Verisilicon Design Service**

**2016/10/26 v0.2**

This document is to define the design quality for projects ship to Verisilicon Design Implementation.

Table of Contents

[Design planning guide 3](#_Toc465258664)

[RTL design guide 5](#_Toc465258665)

[Constrains design guide 6](#_Toc465258666)

[Synthesis guideline 8](#_Toc465258667)

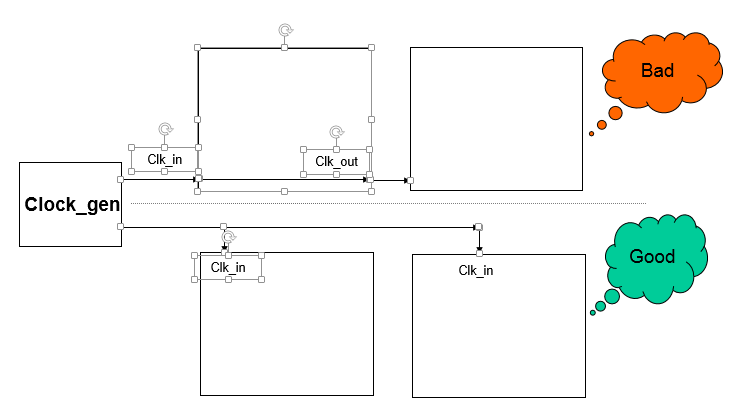
[Netlist design guide 10](#_Toc465258668)

[DFT Test Rule 11](#_Toc465258669)

[Design Data Check in Guidelines 19](#_Toc465258670)

# Design planning guide

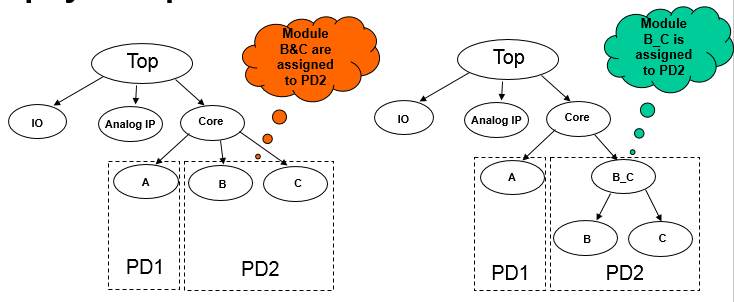
1. Logic hierarchical level recommendation is to less than 15. Design gate count limitation recommendation for each block is 150M instances (better within 100M). And less for Top to get better chip floorplan.
2. Put the high speed/critical path in the same module and avoid the glue logic between the modules, especially the modes which contain the critical paths.
3. Every module gate size must be in a reasonable scale. The gate counts are better within 200K in a module, for synthesis tool limitation and better synthesis QoR.
4. Isolation necessary asynchronous logic into separate blocks, for better timing closure.
5. Clock definition point (either defined by create\_clock or create\_generated\_clock) should be placed at the same module (clock generated module) as well as to put the set/reset root to the reset generated module. A detail document on clock/reset generation module needed for implementation.
6. Keep only on logic module for each power domain or hierarchical partition. Two or more hierarchical modules can’t be assigned to one power domain or hierarchical partition.
7. Make a good partition between logics in order to get reasonable block interface signals. No inter-cross path, snake path, feed through pins and synchronous clock outputs in a partition.



Partition I/O signals should not be too tight for a physical floorplan shape. Leave enough pin space for signals in case of top routing congestion.

1. If low power design, all low power design blocks must be with UPF/CPF.

Only one module is assigned on one power domain or physical partition. No power domain nested. If two modules are in same power domain, please group the two modules as one, like B\_C.



There needs a full chip UPF/CPF for low power check signoff. Block UPF/CPF must be consistent with the full chip one.

1. Try to make the power management simple. Like not to use retention register; less signal crosstalk exist between power domains.

A dedicated Power controller module for power management is recommended. Or control signals from port are OK as well.

Retention registers can be replaced into Design Compiler with UPF. Isolator and level shifter can be inserted in logic synthesis tool and can also be inserted in ICC. Power switch cell can be inserted by ICC with UPF.

Normally:

Isolators are placed in “ON” domain.

Low-to-High level shifters are placed into low voltage domain.

High-to-Low level shifters are not necessary.

# RTL design guide

1. Lint check clean. Like: spyglass.
2. Bus order in MSB->LSB, same for Netlist-in phase.
3. Don’t touch list must be provided at forehead to void them to be removed during synthesis/PR flow. Like: the spare gate and the delay chain and we suggest the customer add the spare gates after PR trail run. Do not connect clock to the spare gates DFF, which will reduce the CTS quality.
4. Don’t use list must be provided at forehead to void them to be used during synthesis/PR flow. Like: ECO usage spare cell and special function cells. Wire/module/reg/cell name should reflect their function and usage, and their name should not too long (not exceed 64 word) and no special character ( obey the verilog naming rule)
5. The top level port should be the single bit if possible, the bus port will bring troubles in the implementation stage.
6. Pad cell should be group in a same module (Pad ring module). Directly instantiate pad cell at top level will lead a quite “dirty” netlist after synthesis and implementation. A pin\_assignment document is reserved for PAD ring design.
7. Use If-ELSE for 2-to-1 multiplexers, and CASE for n-to-1 multiplexers (n> 2), use IF-ELSEIF for priority encoders
8. Use CASE with //synopsys parallel\_case when conditions are mutually exclusive

Use CASE with //synopsys full\_case when not all conditions are specified

Use CASE with //synopsys full\_case paralle\_case for one hot finite state machines

1. Use the parameter to define state values for FSM

Use CASE with //synopsys full\_case paralle\_case to describe FSM, while not to use default unless recovery state is desired

1. Clock gating code recommendation.

Synthesis Tool will not inferred the registers whose clock have been gated:

assign GateClk = EN & Clkin;

always @( posedge GateClk ) begin

RegA <= B

end

If you want tool to use clock gate cell to gate RegA, you should write as following:

always @( posedge GateClk ) begin

if (EN) begin

RegA <= B

end

end

Minimize the level of clock gating. Strongly recommend less than 2.

1. Use parenthesis to guide synthesis

# Constrains design guide

1. SDC must match the related design data (RTL/Netlist).
2. All items of check\_timing, like 'unconstrained paths', 'no\_input\_delay', 'no\_driving\_cell', 'unconstrained\_endpoints', 'unexpandable\_clocks', 'no\_clock', 'loops', etc. are confirmed by designer.
3. Any timing exception, like set\_false\_path, must be fully verified and checked.
4. STA can’t cover the asynchronous path. Customer has to cover the potential timing, glitch, noise risk.
5. Clock should be defined on a leaf pin, in implementation SDC, and the clock is better defined on an output leaf pin. And set\_dont\_touch on related cells to avoid of removing by EDA tool optimization.
6. The synchronize clock group (means there are true path between the different clocks), the period of them should have multiple relationships, for example clock A and clock B are two clock domain, and there have the path from A to B, so if the A clock is 300M and the B clock is 150M, then we should defined the A clock to be 3.3ns and B clocks period should be 3.3 x 2 = 6.6ns but not 6.67ns (1000/150).
7. All the input/out port which have timing requirement should been defined the input/output delay as well as the driven strength/input transition and output load. We will default think that the ports without constraints means there have no corresponding requirement.
8. The input delay or output delay setting might be different before and after CTS, after CTS the network latency of the clock tree should be taken into account.
9. For some path is DDR or DDR like, which need the delay of this group should be a designated scale under min max condition and the delay skew of the group should be also in a quite small scale. Customer should be inform Verisilicon before the PR trail run, for this kind of path can’t not be full constrained by SDC. Dedicated tcl scripts should be provided to check this kind of timing.
10. Please don’t define too much clock at the same point by using the –add option of create\_clock or create\_generated\_clock, for it will decreased the tools efficiency to the timing analysis/optimization. Try to merge some clock definition.
11. Try not to use set\_max/min\_delay in SDC if possible, for this command will break normal synchronous timing path and bring SDF generation issue.
12. The generated clock should be defined at the right point where it is it’s real generated point, for example ( see FIG 1) the generated divided clock have a real root at QN(Green point), but for some reasons this generated clock defined at the end gate’s Z pin ( Red point ), this will lead the clock gating check at the and gate to be lost.



Fig 1

1. The SDC for implementation should not be the synthesis SDC, which is directly generated by using write\_sdc after synthesis. A manually written SDC based on the synthesis SDC are request for design implementation. Accurate clock generated description, no over constraints on frequency or uncertainty.

# Synthesis guideline

1. Group the critical path use the group\_path command, and try to set high weight for them to pay higher synthesis effort for this group
2. Use set\_critical\_range command to reduce the TNS (total negative slack ), and which can let the DC to optimize the negative slack path which is not the worst slack path, and this can help you figure out some path that really can’t be meet the constraints ( normally these path always to be a false path )
3. Used the set\_dont\_use command to forbid the DC to use the some lib cell during synthesis. And flowing cell should be avoid, large driven strength above 20X and small strength below 1X , clock cell which normally with the lib cell name \*CLK\*, and some complicate sequential cell JK , or DFF cell which have enable pin or DFF cell have a select pin to select data (A or B) to be loaded ( Normally this complicated DFF will take negative effect on timing optimization during synthesis)
4. Use the set\_clock\_gating\_style defined the max fanout of single clock gating cell, we suggest that the fanout do not exceed 32, and also set the minimum\_bitwidth option to constraint the minimum registers bus width number to be gated , we suggest the minimum\_bit\_width is 4
5. We strongly recommend the clock/Reset generated module to build up by manual instantiate the gate, for when you do synthesis you can defined the clock or generated clock right on the leaf pin and the SDC which generated after synthesis will not bring to much difference with at the SDC you used in synthesis. If you plan to synthesis the clock /reset generated module, at least we suggest the clock switch cell should be manually instantiated by using a mux cell, for it will be easier to debug clock timing during implementation
6. Timing margin recommendation.

* For 130nm or above,

“Zero-WLM”: 20% clock period + sign-off clock uncertainty (300ps).

* For others, (if very high speed clock, case by case)

“Zero-WLM”: 30% clock period + sign-off clock uncertainty.

“DC-T/G”: 300ps (OCV+SI) + sign-off clock uncertainty.

“DC-G” is preferred for better correlation and lower congestion.

The timing report generated by synthesis tool are needed to review the timing quality. Verisilicon will run pre-layout STA for timing margin check. It is to evaluate the netlist to be qualified or not for implementation.

1. Use “set\_fix\_multiple\_port\_nets –all –buffer\_constants” to avoid the assign statement in netlist
2. Use “change\_names –rules verilog –hierarchial” to avoid the special character used in netlist.
3. Because DC will remove the redundancy logic which takes no effect for the design function, so if you have forget connected some modules’ pin and leave them unconnected will lead the tool remove the logic which you really want to keep them. To avoid that We suggest you should take notice to the Flip-Flop cells that have been removed by review the log file or in SVF file
4. Use “compile\_ultra –noautoungroup” command to do the synthesis to get better synthesis QoR; if scan is wanted, use “compile\_ultra –noautoungroup –scan” to considerate the scan replace effect during synthesis stage. And use “set auto\_disable\_drc\_nets –scan” to make scan related signals (Scan enable or Test mode) ideal by DC.
5. Recommended setting for synthesis

set hdlin\_no\_group\_register              TRUE

set enable\_recovery\_removal\_arcs         TRUE

set power\_preserve\_rtl\_hier\_names        TRUE

set power\_hdlc\_do\_not\_split\_cg\_cells     TRUE

set power\_do\_not\_size\_icg\_cells          FALSE

set compile\_new\_optimization             TRUE

set compile\_seqmap\_propagate\_constants   FALSE

set timing\_non\_unate\_clock\_compatibility TRUE

set compile\_delete\_unloaded\_sequential\_cells TRUE

set compile\_ultra\_ungroup\_dw  TRUE

set high\_fanout\_net\_pin\_capacitance  0

set case\_analysis\_with\_logic\_constants TRUE

set timing\_input\_port\_default\_clock FALSE

1. For some design having some critical data path, The efficient way is to explicit instruct the DC to use the high speed arithmetic designware implement, for example using set\_impl\_priority –priority 0 standard.sldb/DW01\_add/cla in DC to use the cla as the adder implementation.

# Netlist design guide

1. No assign statement.
2. No floating input pins (both hierarchical pin and leaf input pin).
3. No special characters in gate-level netlist.
4. No mult-driven nets; Port/Net name cannot be as same as instance name.
5. Be careful on latch code, no check on the latches in the design, Verisilicon will take any latches in the netlist are intentional added
6. Constant bus width can’t exceed 32 bit in netlist, for example 64’b0000000……., PR tool have some limitation to identify such long constant bus in netlist
7. No empty module in the netlist, which is a IP and have a corresponding liberty linked (Normally customer some time add some empty module in their design, when there are no liberty available and they forget to remove the empty module when the liberty is ready)
8. Multi-driven net is forbidden, which will bring troubles both in simulation and implementation
9. No large driven strength buffer/inverter (above 20X) or low driven strength cell (below 1X); No complicated sequential cells but to use the normal DFF cell and latch.
10. Netlist and SDC for implementation must be checked before release. Both syntax and timing margin should follow Verislicon’s design guideline.
11. If low power design, released Netlist should be with low power cell inserted and pass the low power check with released CPF/UPF.

# DFT Test Rule

1. A test plan for whole chip need to be made before DFT insertion. Verisilicon DFT engineer can help on DFT specification setup. Mode decode and pin share plan are necessary. This can also be merged into the pin\_assignment document.
2. At-speed test are recommended for 65nm and below, while IDDQ are not. Scan DC test coverage 95%+, AC test coverage 80%+.
3. Detail DFT request should be provided, or Verisilicon will insert MBIST, SCAN and BSD as the default DFT items, Test clock 10Mhz-30Mhz, Chain length <= 1000, No any IP related test implemented. Verisilicon will do nothing on any analog IP test.
4. It is recommended to add the bypass logic (see fig2) for the scan clock/reset with the functional clock/reset in at the design stage to achieve high test coverage, for example , you should write the code as flowing

assign Clock = ScanMode? ScanClock : FunctionClock;

assign Reset = ScanMode? ScanReset: FunctionReset;

always (@ posedge/negedge Clock or Reset) begin

….

….

end



Fig 2

1. There are several basic signals should be used for Scan test, like : Scan mode, Scan enable, Scan Reset, Scan clock, Scan in, Scan out. Customer need provide the corresponding ports that can reused as the scan signal port, and provide the corresponding “hookup pin” in the design (See fig 3 Red character is the hook pin ). So a pin mux module which is used for the test signal generated module is needed. For detail signal share, Verisilicon can provide DFT spec. normally, for an at-speed DFT, 10-15 more inputs than outputs. E.g. 40 scan-out needs about 55 input scan signals. These signals list can be used for the pin mux module design.

Inputs: Scan\_mode, Scan\_enable, Scan\_Reset, Scan\_clock, Scan\_in[MSB1:LSB1]

Scan\_compression (if compression needed)

Scan\_set (if both reset and set register used in design)

Occ\_bypass, Occ\_reset (if at\_speed test OCC used)

ATE\_clock[MSB2:LSB2] (for async OCC clock groups)

Outputs: Scan\_out[MSB1:LSB1]

 Fig3

1. We recommended that all the bidirectional port under scan test (include normal stuck at test, transition test, or path delay test) should have a fixed direction whether in or out. scan in/ scan clock/scan reset port to be fixed to input and scan out fixed to out, and the left ports which haven’t used for scan can fixed either input or output
2. Remove the pullup/pulldown logic under DFT modes if possible.
3. Try to reuse the port which used as a clock/ reset in mission mode to reused as a scan clock/reset, and if you have to use the signal port to be used as a scan clock/reset, please do as the Fig 4 to avoid the clock as data violation



Fig 4

1. Use the Pre-observer ICG cell when do the clock gating to improve the test coverage. If special clock gating cell which must fixed to pass at-speed clocks, please list them so that DFT won’t use scan\_enable signal to control it.
2. All the flip flop should be included in the scan chains to obtain high test converge, and add test wraps to IP/memory or the modules which haven’t included in scan chains (see Fig 5)



Fig 5

1. Use scan mode signal to gate some logic will reduce the test coverage and remove any gate logic that no need to be gated under scan mode
2. For at speed test/transition test, details on how to share and reuse FUNCTION PLL and clock generator are needed, clock diagram and clock start up sequence are to be released.
3. There are one-to-one or more-to one relationship between the function clock and at speed test clock. You should do the planning of which clock domain under the functional mode to be test by which at speed clock. For example (see fig 6), there are two functional clock A and B with are the same frequency 200M and they are talked each other under the functional mode then you should assign both of them to tested by At speed clock 1, but for functional clock C which is 133M and asynchronies to function clock A and B, then you can assign another at speed clock 2 which is 133M test frequency.

Fig 6

1. There are two different at speed clock source, one is directly from pad (external clock), another is from OCC (on chip clock, whose mast source is from PLL), for frequency below 200M we can both used the external clock and OCC clock, for clock frequency exceed 200M, we have to use OCC only. Please see the Fig 7 for the application



Fig 7

1. For MBIST insertion, Mentor Tessent MBIST is the popular flow, TAP controller is used for top integration. 5 standard signals under MBIST mode: 4 inputs TCK, TRST, TMS, TDI and 1 output TDO. If a non-at-speed mode wanted, extra 2 input signals are needed for low speed switch and low speed clock input. If diagnostic needed, one more diagnose output signal needed. These signals list can be used for the pin mux module design.

Inputs: TCK, TRST, TMS, TDI

MB\_LOWSPEED, MB\_LOWCLK (if low speed MBIST wanted)

Outputs: TDO

MB\_DIAGOUT (if diagnose wanted)

1. The default MBIST test algorithm is SMARCHCHKBVCD for SRAM, ReadOnly for ROM if no special requirement for memory test faults type. Memory list must be provided for memory type sorting and MBIST controller grouping.
2. For at-speed MBIST, memory clock will reuse the function clock tree, please add MBIST clock mux just like at-speed SCAN clock mux.
3. For BSD, JTAG TAP has 5 standard signals. It is recommended to use dedicated IO for BSD insertion, if not, pin mux logic with hookup\_pin mapping needed for the 5 singals. These signals list can be used for the pin mux module design.

Inputs: TCK, TRST, TMS, TDI

outputs: TDO

# Design Data Check in Guidelines

|  |  |  |  |
| --- | --- | --- | --- |
| Check In Items | | Y/N | Remarks |
| Library-in  **🞎** | Full libraries/IP usage table including library name and version |  |  |
| Unified Database directory structure for easy collecting and setup environment, better to have description documents or notes |  |  |
| STD/IO/MEM/IP full DK or TK(LEF, Verilog, LIB, DFT related model, GDS, CDL) |  |  |
| STD/IO libraries ensure ccs model and full signoff corners, better to have milkyway database. |  |  |
| Implementation guideline documents for libraries/IP from Vendor |  |  |
| Process design guide, reference flow. |  |  |
| Process signoff guide for timing, power and EM/IR. |  |  |
| Integration guideline documents and DFT related documents for special soft IP or block from owner, like CPU(A53), GPU(Mali820), VE |  |  |
| ICC related file:  Techfile  layer map file  antenna rule file  tluplus |  | Can be multi-file for different STD library |
| Innovus related file:  Tech LEF  Layer map file  Antenna LEF  QRC techfile  QRC related file:  QRC techfile  Layer map file |  |  |
| starRCXT related file:  itf  nxtgrd  Layer map file |  |  |
| Virtuoso releate file:  Techfile  display file  map file |  |  |
| Redhawk related file:  techfile with EM rule on TT85  apl file |  |  |
| Calibre related file:  DRC runset  ANTENNA runset  LVS runset  Dummy runset  Bond runset  PERC runset |  |  |
| Design  Document | Design data flow diagram |  |  |
| Design floorplan reference (if previous version exists) |  |  |
| Pin assignment form (if bump map exists, please consider the RDL routing on pin assignment) |  | Verisilicon template |
| Detail Clock/Reset design structure diagram (match the SDC and Design code) |  |  |
| Low power design structure diagram |  |  |
| Memory instance list and configuration file |  |  |
| DFT detail request document |  | Test plan |
| DFT mode pin share plan and document(can be merged into pin assignment) |  |  |
| PLL and Clock generation reuse in DFT mode document (if at\_speed DFT request) |  |  |
| Customer sign off criteria requirement |  |  |
| EDA Tool Version requirement |  |  |
| Special manual implementation items(like spare cells) |  |  |
| Don’t use list |  | Can be tcl format |
| Don’t touch list |  | Can be tcl format |
| Design  RTL-in  **🞎** | RTL codes(Verilog/VHDL) with file list  \*\* followed RTL design guide and Design planning guide |  | Pass lint check |
| SDC for synthesis matches the RTL codes  \*\* followed Constrains design guide |  | Pass syntax check |
| UPF/CPF for power intent (if lowpower design)  \*\* followed Design planning guide |  | Pass low power check with RTL code |
| With good timing margin with the synthesis library |  | See Netlist-in Timing recommendation |
| With good DFT testability in coding style |  | Pass DFT DRC rules |
| Design  Netlist-in  **🞎** | Full Integrated Netlist for top and each block  \*\* followed Netlist design guide |  | Pass syntax and pre-timing check |
| SDC for design implementation matches the netlist  \*\* followed Constrains design guide |  | Pass syntax and pre-timing check |
| UPF/CPF for power intent (if lowpower design)  \*\* followed Design planning guide |  | Pass low power check with netlist |
| Synthesis timing reports.  For 130nm or above,  “Zero-WLM” 20% clock period + sign-off clock uncertainty (300ps).  For others,  “Zero-WLM” 30% clock period + sign-off clock uncertainty. (if very high speed clock, case by case)  “DC-T/G”: 300ps (OCV+SI) + sign-off clock uncertainty. (“DC-G” is preferred for better correlation and lower congestion for 28nm and below) |  | Timing margin recommendation |
| Physical constraints scripts (if have the request) |  |  |
| Asynchronies or Skew timing check scripts (if have the request) |  | Like DDR skew request |
| Function typical work period simulation VCD (if dynamic IR-drop needed) |  | Toggle rate for implementation must be pessimistic than VCD on power |
|  |  |  |  |
|  |  |  |  |