

CHECKLIST FOR SUCCESSFUL SYNTHESIS

The enclosed checklist is intended to aid the designer in developing a successful synthesis methodology. It is not intended to replace the Synopsys documentation, but rather to supplement it. Each activity has an associated reference document where the designer can access more information about the topic or command.

The document reference should be considered a starting point into the documentation. Often the topic may be discussed in more than one place within the Synopsys On-line Documents. The reader is urged to do an "index search" to find additional information. An index search can be done using an Adobe Acrobat reader with search capabilities.

Several Application Notes are referenced. Many of these are available on Solvit (SolvNET) or from your local Applications Consultant.

REVISION 2.0
DECEMBER 15, 1999

AUTHOR:

JASON WARE – SENIOR APPLICATIONS CONSULTANT

SYNOPSYS™

Check List For Successful Synthesis

Planning Stage

✓	Action	Document	Chap
	Consider Synopsys training courses	www.synopsys.com(education)	
	Update software and licenses	Installation User Guide	1
	Access online documents	\$SYNOPSYS/doc/online/top.pdf	
	Setup design directory structures	DC User Guide	3
	Setup .synopsys_dc.setup file	DC Ref: Fundamentals	4
	Define and evaluate link, target and symbol library	DC Ref: Fundamentals	5
	Define proper DesignWare libraries	DC Ref: Fundamentals	5
	Include “*” at beginning of link_library	DC Ref: Fundamentals	5
	Define a naming convention - no reserved words	VHDL/Verilog Ref Man	AppC
	Develop revision control scheme	DC User Guide	3
	Define meaningful hierarchy partitions	DC User Guide	3
	Consider test case design to test flow	DC User Guide	2
	Read Command Quick Reference Manual	Synthesis Quick Ref.	All
	Define libraries, filenames, etc. as variables at top of synthesis scripts	DC Command-Line Interf.	3

Design Environment and Constraints

✓	Action	Document	Chap
	Specify wireload models	DC Ref: Constraints	3
	Specify design rule constraints	DC Ref: Constraints	3
	Define clocks	DC User Guide	7
	Define common base period for clocks	DC User Guide	7
	Use clock_uncertainty for clocks with slightly different frequencies	DC User Guide	7
	Define interconnect mode	DC User Guide	7
	Define operating conditions	DC Ref: Constraints	3
	Define drive strength	DC User Guide	5
	Define output port loads	DC User Guide	5
	Model clock trees	DC User Guide	7
	Define virtual clocks if needed.	DC Ref: Constraints	4
	Specify input/output constraints.	DC User Guide	5
	Define realistic area goals	DC Ref: Constraints	2
	Define timing exceptions	DC User Guide	7
	Constrain for glitch free gated clocks: set_clock_gating_check	DC Ref: Constraints	4

Partitioning

✓	Action	Document	Chap
	Eliminate unnecessary hierarchy	DC User Guide	3
	Ensure top level block contains only I/O and hierarchy blocks	DC User Guide	3
	Separate core logic, pads and clock divider logic.	DC User Guide	3
	Eliminate glue logic at the top level	DC User Guide	3
	Eliminate hierarchy in combinational paths	DC User Guide	3
	Register top-level outputs of compile blocks	DC User Guide	3
	Partition by design goal, separate speed and area critical blocks	DC User Guide	3
	Balance block size with run times	DC User Guide	3

Proper Coding Styles For Synthesis

✓	Action	Document	Chap
	Make flip-flop clk and async. pins controllable	DC User Guide	3
	Never use high-impedance values in conditional statement	DC User Guide	3
	Be aware of prioritization i.e. if/else vs. case branches	Guide to HDL Coding Style	2
	“If” without “else” generates latch in comb. block	DC User Guide	3
	Use case statement for more than 3 if-else statements	Guide to HDL Coding Style	2
	“Define” / “constants” are global - put in separate file	DC User Guide	3
	“Parameters” / “generics” are local	DC User Guide	3
	Structure code to take advantage of resource sharing	Guide to HDL Coding Style	5
	Use parenthesis to indicate precedence in an expression	DC User Guide	3
	Do not use global references within a function	DC User Guide	3
	Verilog: Be aware of difference between simulation and synthesis for blocking vs. non-blocking assignments	HDL Compiler Ref Man	5
	VHDL: Only signals within sequential processes. Variables as much as possible within combinatorial processes (For simulation performance)	VHDL Comp Ref Man	6
	Verilog: Only non-blocking assignments within sequential always@ blocks. Blocking assignments as much as possible within combinational always@ blocks	HDL Compiler Ref Man	5
	Completely specify sensitivity lists	DC User Guide	5
	Be careful with for loops (Implies slow cascade of logic)	Solvit: MISC-247579	
	Consider use of advanced DesignWare Foundation components for speed and area critical designs	DF Foundation Data Book	All

Design For Test

✓	Action	Document	Chap
	Perform one-pass scan, i.e. compile -scan	TC Reference Guide	5
	Set scan configurations	Scan Synth User Guide	2
	Use set_scan_element false rather than dont_touch	Scan Synth User Guide	2
	Avoid combinational feedback loops	Scan Synth User Guide	4
	Asynchronous preset and clears should be controllable	Scan Synth User Guide	3
	Consider multiple scan chains to reduce vector count	Scan Synth User Guide	3
	Run check_test before and after insert_scan	Scan Synth User Guide	4
	Examine errors, i.e. “help TEST-XXXX”	Scan Synth User Guide	4
	Consider BSD Compiler to insert JTAG	BSD Compiler User Guide	ALL

Design Exploration

✓	Action	Document	Chap
	Execute quick preliminary synthesis in parallel with functional RTL simulation	DC User Guide	2
	Define time budgets when performing bottom up compile	Design Budget User Guide	All
	Develop scripts	DC Command-Line Interf.	8,10
	Examine constraints: report_timing_requirements	DC User Guide	2
	Resolve multiple instances	DC User Guide	8
	Reduce timing violations to less than 10% over constraints	DC User Guide	3
	Eliminate all design rule violations	DC User Guide	8
	Eliminate all test rule violations	Scan Synth User Guide	4
	Explore and understand different compile options	DC User Guide	8

Design Synthesis

✓	Action	Document	Chap
	Adopt top-down methods where possible	DC Ref: Opt and Timing	2
	Always start with default compile	DC Ref: Opt and Timing	3
	Fix setup / hold / design rule violations	DC Reference: Constraints	4
	Add extra cells for later ECO capability	ECO Compiler Reference	5
	Understand the limits of boundary optimization	DC User Guide	8
	Fix violations greater than 10% at the code level	DC User Guide	3
	Consider register repositioning to meet timing	DC Reference: Optimization	5
	Ungroup DesignWare parts to meet timing and area goals	DC User Guide	AppA

Design Analysis

✓	Action	Document	Chap
	Check report files for latch inference	Guide to HDL Coding Style	5
	Look for implied dont_touch attributes	Solvit: METH-1319	
	Look for remaining GTECH cells after mapping	DC User Guide	9
	Examine DesignWare resource implementation	DC User Guide	9
	Perform timing analysis to pinpoint problems	DC User Guide	9
	Divide timing reports into inputs, outputs and register-to-register groups	DC Ref: Opt and Timing	12
	Spot large delays on cells in timing report and identify cause	DC User Guide	9
	Trace problems back to RTL	DC User Guide	9
	Modify hierarchy if necessary	DC User Guide	3

Links to Layout

✓	Action	Document	Chap
	Qualify back end tools ahead of time.	Floor Plan Man. User Guide	2
	Consider trial floorplan and/or layout	Floor Plan Man. User Guide	1
	Decide on Bottom up or top down flow	Floor Plan Man. User Guide	8
	Use report_qor to create a detailed design report including the number of instances in a design	Floor Plan Man. User Guide	2
	Use PDEF 2.0 rather than 1.0 for all but the simplest designs.	Floor Plan Man. User Guide	3
	Consult “lessons learned” section of app note	FPM Secrets for Success	12
	Realize an FPM flow is sufficient to handle errors up to 20% of the clock period.	FPM Secrets for Success	9
	Avoid setting constraints that result in negative paths.	Floor Plan Man. User Guide	2
	Forward annotate constraints to layout tool	Floor Plan Man. User Guide	2
	Be aware of command changes with 1999.05 release.	FPM Secrets for Success	All
	Check timing report for “*” after back annotation	Floor Plan Man. User Guide	5
	Create custom wireload models	Floor Plan Man. User Guide	4
	Examine the CWLM standard formatted table to determine the quality of the model	FPM Secrets for Success	6
	Consider a two-stage compile (fix design rules first then timing).	FPM Secrets for Success	9
	Recompile starting with reoptimize_design –map_effort medium (the default).	FPM Secrets for Success	9
	Continue with the sizing, buffer insert/remove and pin swapping options until constraints are met.	FPM Secrets for Success	9
	For large errors the –map_effort high may be used, however at that point consider RTL/synthesis changes.	FPM Secrets for Success	9
	Examine post layout timing report: check_design -post_layout	Floor Plan Man. User Guide	5

Problem Resolution, IN ORDER...

	Consult documentation
	Look for Solvit article
	Consult internal ACE
	Produce test case for handoff.
	Contact support center
	Consult local Application Consultant
	Consider additional on-site consulting