

IC Compiler 2010.03 Incremental Training

Timing Updates

Agenda



1. AOCV Support
2. Clock Arnoldi Flow Changes
3. Pre-route CCS Support
4. Input Port Clock Changes
5. Case Analysis Sequential Propagation

AOCV Support

Introduction

- Traditional on-chip variation (OCV) approach can be pessimistic
 - Uses `set_timing_derate` to apply global derating factors on design/library/data/clock/instance
 - Could result in pessimism and over-design in lower technology nodes
- Advanced OCV for best accuracy and QoR
 - Uses context-specific derate values, excessive design margins are reduced and fewer timing violations are seen
 - Produces better accuracy and synthesis QoR by reducing unnecessary over-design

	IC Compiler Traditional OCV	IC Compiler Advanced OCV 2009.06-SP3
Approach	Global derate for early/late/min/max/clock/data...	Clock path derate as a function of logic depth
Benefit	Easy set-up using linear delay scaling factors	Intermediate step to statistical; Provides first-order model of variation
Challenge	Safe, but pessimistic; becomes overloaded to model many effects	Requires silicon data from foundries or in-house fabs

AOCV Support

IC Compiler solution

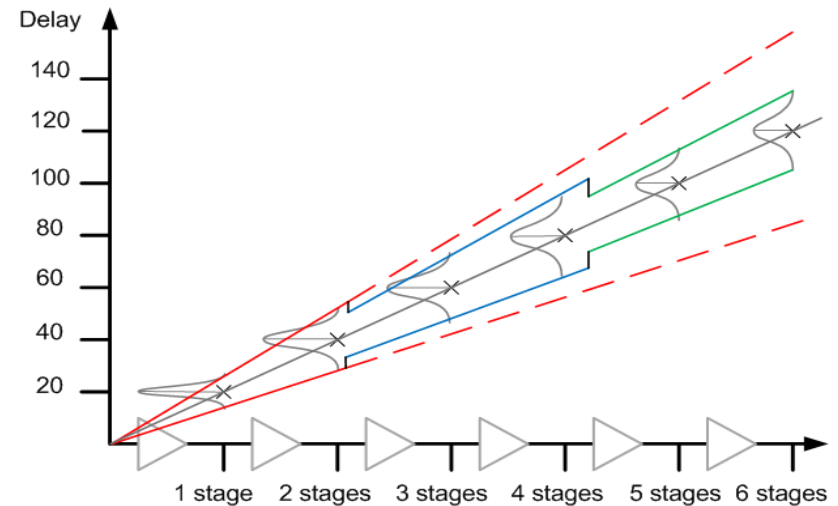
- Calculate and apply variable derate factors that model process variations more closely than traditional constant derating
- Random variation effects from clock path handled via derate tables
 - Depth used to index random component of variation in AOCV derate table
 - Random variation as a function of logical path depth
 - Advanced OCV derates are computed and applied only to the non-common portion of the clock networks
 - The data network uses normal OCV derates
 - Compatible to PT Graph Based clock network only Advance OCV mode

```
pt_shell> set timing_aocvm_analysis_mode clock_network_only
```

AOCV Support in IC Compiler

Background

- More accurate modeling for advanced processes
 - Deep paths exhibit less variation than short ones
 - Derate based on logic depth
- Already supported in post-route flow through PrimeTime exact link
 - In TSMC Ref Flow 9.0
- Extended to post CTS for hold fixing, qualified in TSMC Ref Flow 10.0
 - AOCV derates used on clock network
 - Global derates used on data paths
 - Uses PrimeTime derating tables
 - Tables can be generated using statistical HSPICE



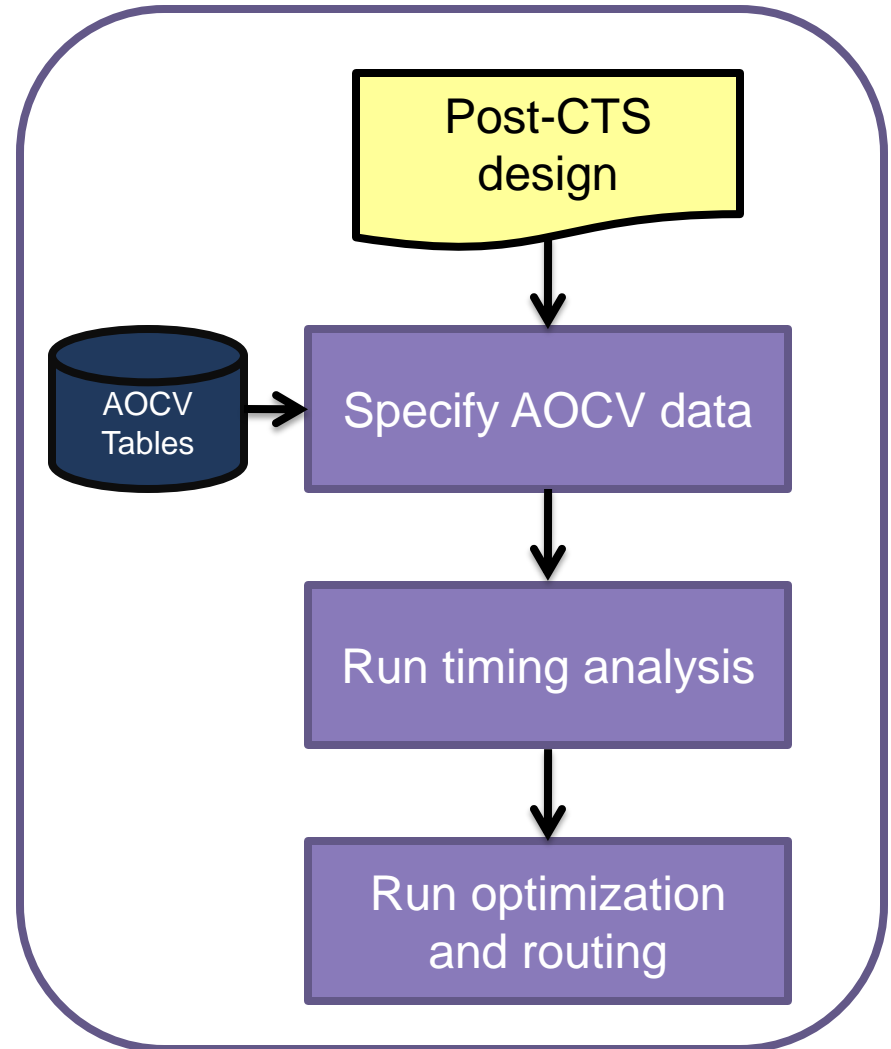
Depth	1	2	3	4	5	6
Derate	1.2	1.2	1.15	1.15	1.08	1.08

- Requires Galaxy Advanced Technologies license

AOCV Support in IC Compiler

Flow Setup And Usage Model

- New commands
 - `read_aocvm`
 - Reads in AOCV tables for design, hierarchical cells, or library cells
 - `set_aocvm_coefficient`
 - Defines an additional path depth increment for leaf cells or library cells
 - `report_aocvm`
 - Reports specified AOCV data and computed derates
 - `remove_aocvm`
 - Removes AOCV data specified
- New variables
 - `timing_aocvm_enable_analysis`
 - Enables AOCV analysis



AOCV Support

TCL Script And Annotation Report

```
clock_opt -only_cts
```

```
set timing_aocvm_enable_analysis true  
read_aocvm aocv_table  
report_aocvm  
report_timing -path full_clock_expanded -derate
```

```
psynopt  
report_timing -path full_clock_expanded -derate
```

```
route_opt  
report_timing  
-path full_clock_expanded  
-derate
```

	Total	Fully annotated	Partially annotated	Not annotated
Leaf cells	7908	7908	0	0
Nets	8036	8036	0	0
	15944	15944	0	0

Cell

Coefficients

CTS_clk_delay10	2.0000
CTS_clk_delay12	1.9800
CTS_clk_delay13	2.0000
CTS_clk_delay111	2.0300

AOCV Support

Timing Reports

Advanced OCV

Traditional Constant Derate Approach

Point	Fanout	Derate	Incr	Path
clock clk (rise edge)			0.00	0.00
CLK (in)		1.11	0.00	0.00 r
CLK (net)	1		0.00	0.00 r
CTS_clk_delay13/A (buf1a1)		1.10	0.00 *	0.00 r
CTS_clk_delay13/Y (buf1a1)		1.11	0.11	0.11 r
CTS_clk_delay131 (net)	1		0.00	0.11 r
.....				
buf1a15/A (clk1a15)		1.14	0.01 *	0.84 r
buf1a15/Y (clk1a15)		1.21	0.23	1.08 r
CLK__1 (net)	27		0.00	1.08 r
REG_7/CLK (fdf2a9)		1.18	0.00 *	1.08 r
.....				
N10 (net)	1		0.00	4.92 r
REG_18A/D (fdf1a2)			0.00 *	4.92 r
data arrival time				4.92
clock clk (rise edge)			3.00	3.00
CLK (in)		0.89	0.00	3.00 r
CLK (net)	1		0.00	3.00 r
CTS_clk_delay13/A (buf1a1)		0.91	0.00 *	3.00 r
CTS_clk_delay13/Y (buf1a1)		0.89	0.09	3.09 r
.....				
REG_18A/CLK (fdf1a2)		0.82	0.00 *	3.83 r
REG_18A/CLK (fdf1a2)			0.00	4.03 r
library setup time			-0.10	3.73
data required time				3.73
data required time				3.73
data arrival time				-4.92
slack (VIOLATED)				-1.19

Point	Fanout	Derate	Incr	Path
rise edge)			0.00	0.00
	1	1.25	0.00	0.00 r
elay13/A (buf1a1)		1.25	0.00 *	0.00 r
elay13/Y (buf1a1)		1.25	0.12	0.12 r
elay131 (net)	1	1.25	0.00	0.12 r
elay12/A (buf1a1)		1.25	0.00 *	0.12 r
.....				
K (fdf2a9)		1.25	0.00 *	1.19 r
K (fdf2a9)			0.00	1.19 r
fdf2a9)			0.54	1.73 f
.....				
	1		0.00	5.03 r
D (fdf1a2)			0.00 *	5.03 r
time				5.03
rise edge)			3.00	3.00
		0.75	0.00	3.00 r
	1	0.75	0.00	3.00 r
t)	23	0.75	0.00	3.71 r
CLK (fdf1a2)		0.75	0.00 *	3.71 r
CLK (fdf1a2)			0.00	3.91 r
p time			-0.10	3.61
ed time				3.61
ed time				3.61
time				-5.03
ATED)				-1.42

Agenda

1. AOCV Support



2. Clock Arnoldi Flow Changes

3. Pre-route CCS Support

4. Input Port Clock Changes

5. Case Analysis Sequential Propagation

Clock Arnoldi Flow Changes


Introduction

- Overview
 - Previous flow was complex, and had limitations (e.g. resulted in timing inconsistency in and after optimization)
 - Non-MCMM and MCMM designs were not handled in a consistent manner
- UI
 - No new command/variable
 - `set_delay_calculation -clock_arnoldi` turns on clock_arnoldi
 - Run `route_zrt_group -all_clock_nets \ -reuse_existing_global_route true \ -stop_after_global_route true` and `extract_rc` after optimization to get latest clock skews

Clock Arnoldi Flow Changes

- Old flow
 - Use back-annotation on timing arcs.
 - Changes in pre-route optimization to clock nets takes out net BA, but tool does not have arnoldi data to redo calculation.
- New flow
 - Freezes clock arnoldi skews in pre-route optimization.
 - Command `report_timing -path full_clock_expanded` will not show detailed skew calculation, but use frozen skew value.
- Benefit
 - The flow has been simplified; MCMM handling is consistent

Agenda

1. AOCV Support
2. Clock Arnoldi Flow Changes
-  3. Pre-route CCS Support
4. Input Port Clock Changes
5. Case Analysis Sequential Propagation

Pre-route CCS Support

Introduction

- Overview

- Majority of CCS libraries are CCS/NLDM mixed. Currently DC/ICC uses NLDM in pre-route when it is present.
- Increased usage of CCS libraries and CCS characterization more mature. Users sign-off with CCS in PT.
- Mismatch of NLDM and CCS delays causes qor jump.

- UI

- New variables


- `lib_pin_using_cap_from_ccs` - default to use CCS pincap
- `lib_cell_using_delay_from_ccs` – default to use CCS timing tables

Pre-route CCS Support

Impact

- User Benefit
 - More accurate timing model in pre-route. Better timing convergence with post-route
 - No degradation in runtime/memory/qor
- Flow recommendation
 - Benefit NLDM/CCS mixed libraries.
 - NLDM only libraries are not affected, continue to use NLDM data.
 - CCS only libraries are not affected, continue to use CCS data.
 - Changes affect DC/DCT/ICC pre-route analysis/optimization flows.

Agenda

1. AOCV Support
2. Clock Arnoldi Flow Changes
3. Pre-route CCS Support
-  4. Input Port Clock Changes
5. Case Analysis Sequential Propagation

Input Port Clock Changes

Introduction


- Overview
 - When input ports are not constrained properly by `set_input_delay` command, tools picks a clock from the design to launch input signal.
 - Assumption by the tool can be overly pessimistic and can cause over-constraining, resulting in long runtime and over-optimization.
 - The purpose of this project is to change the default inference of magic port clocks in 2010.03, DC / ICC Timer will not constrain the unconstrained I/O paths.
- UI
 - No new command/variable
 - Variable `timing_input_port_default_clock` will default to false in new release.

Input Port Clock Changes

Impact

- User Benefit
 - New flow avoid over constraining of improperly constrained designs, and reduce runtime.
 - Should not bring significant degradation to runtime/memory/qor.
- Flow recommendation
 - Affects all synthesis tools and flows including all of DC/DCT/ICC.
 - PT `timing_input_port_default_clock` variable defaults to `true`. Users need to change variable in ICC for correlation runs.
 - Variable already in ICC correlation checklist.

Agenda

1. AOCV Support
2. Clock Arnoldi Flow Changes
3. Pre-route CCS Support
4. Input Port Clock Changes
-  5. Case Analysis Sequential Propagation

Case Analysis Sequential Propagation

Introduction

- Overview
 - Current constant propagation stops at sequential cells.
 - New feature allows case analysis to perform on sequential cells and propagates beyond when needed.
 - Could mask out case analysis paths from sequential cells and reduce over-optimization.
- UI
 - No command change
 - New variable: `set_app_var case_analysis_sequential_propagation always/never[default]`
 - Value “always” turns on new feature.

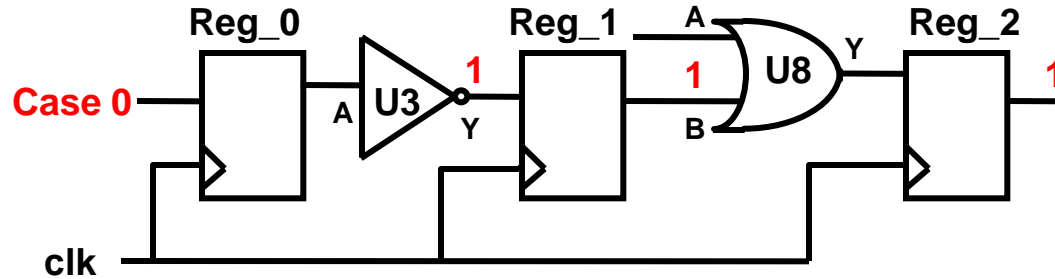
Case Analysis Sequential Propagation

Impact

- User Benefit
 - New feature is consistent with PT capability
 - Can potentially avoid over-optimization on design with case analysis.
 - Should not bring significant degradation to runtime/memory/qor.
- Flow recommendation
 - Feature support in all flows of timing analysis and optimization in DC/DCT/ICC.

Case Analysis Sequential Propagation

Example



Default Flow

report_disable_timing

Flags: c case-analysis
f false net arc
C Conditional arc
l loop breaking
u user-defined

Cell or Port Flag	From	To	Sense	
reg_0	CLK	D	setup_clk_rise	c
reg_0	CLK	D	hold_clk_rise	c

New Flow

report_disable_timing

Cell or Port Flag	From	To	Sense	
reg_0	CLK	D	setup_clk_rise	c
reg_0	CLK	D	hold_clk_rise	c
reg_0	CLK	Q	rising_edge	c
reg_1	CLK	D	setup_clk_rise	c
reg_1	CLK	D	hold_clk_rise	c
reg_1	CLK	Q	rising_edge	c
reg_2	CLK	D	setup_clk_rise	c
reg_2	CLK	D	hold_clk_rise	c
reg_2	CLK	Q	rising_edge	c
data_out				c

Summary of Timing Updates

- AOCV is supported for clock nets
- Clock Arnoldi changes enable consistent timing behavior in/after optimization; Non-MCMM and MCMM designs are likewise handled in a consistent manner.
- CCS support is available at pre-route stage
- ICC timer (like DC) will no longer constrains unconstrained input ports by default
- You can control case analysis sequential propagation for consistent behavior across DC / DCT / ICC / PT

SYNOPSYS®

Predictable Success