Mode Merging: Identification of Mergeable Modes

Subramanyam Sripada

Murthy Palla

Synopsys Inc

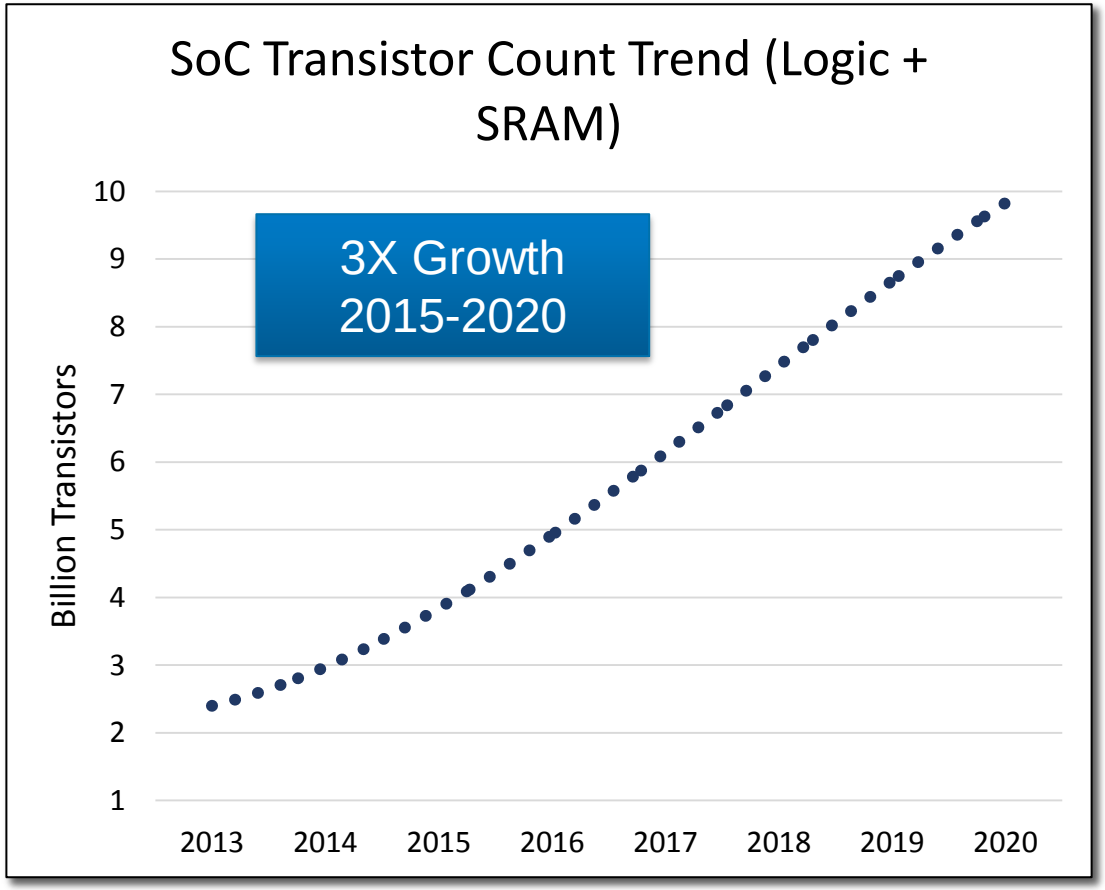
Mar 11, 2016

Agenda

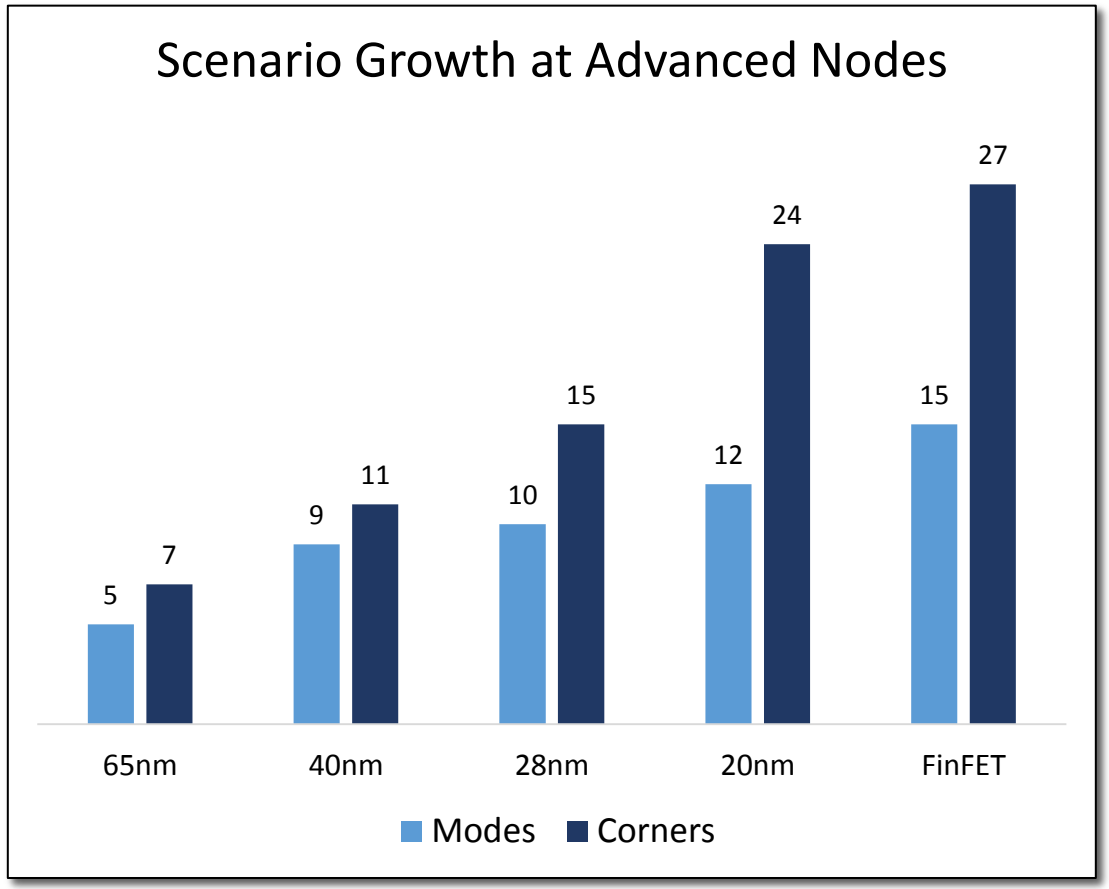
- Motivation
- Background
- Approach
  - Conflicts
- Results

Design/Complexity Projections

An idea of what you can expect



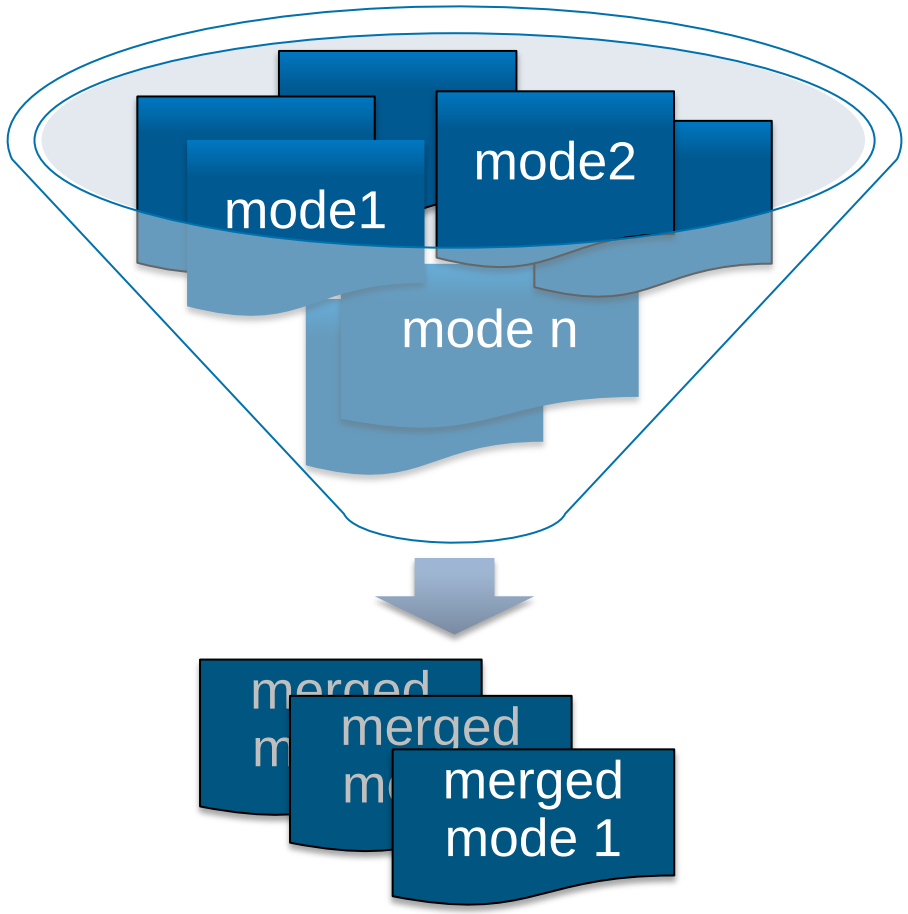
Source: ITRS 2013



Source: Synopsys Customer & Partner Data

SYNOPSYS®

Mode Merging



- It is increasingly common to have large numbers of scenarios for implementation and signoff
- Mode merging helps by collapsing modes where possible
- Analysis with merged modes requires fewer resources

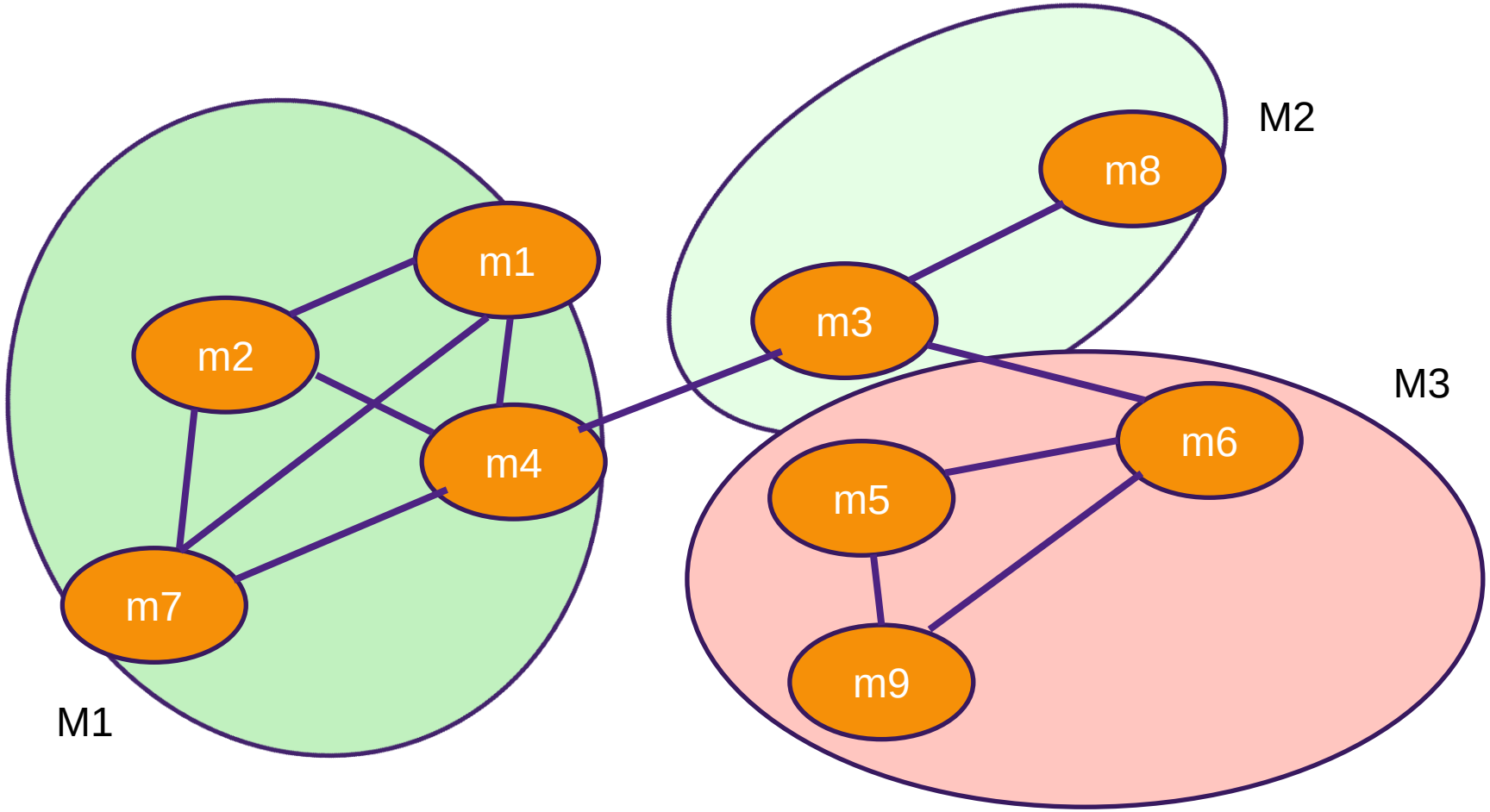
"A Timing Graph Based Approach to Mode Merging", DAC, 2015 describes full approach of merging K mergeable modes to ONE merged mode

SYNOPSYS®

Automatic Mode Merging

- Given N modes, merge them into M modes automatically
  - Determine pair-wise compatibility among modes
    - Can mode A be merged with mode B?
      - If not, can A' be merged with B? (A' exists in theory only – details later)
  - Identify "cliques" in mode compatibility graph
  - Merge all modes in each clique by using an algorithm based on timing graph

Mode Compatibility Graph

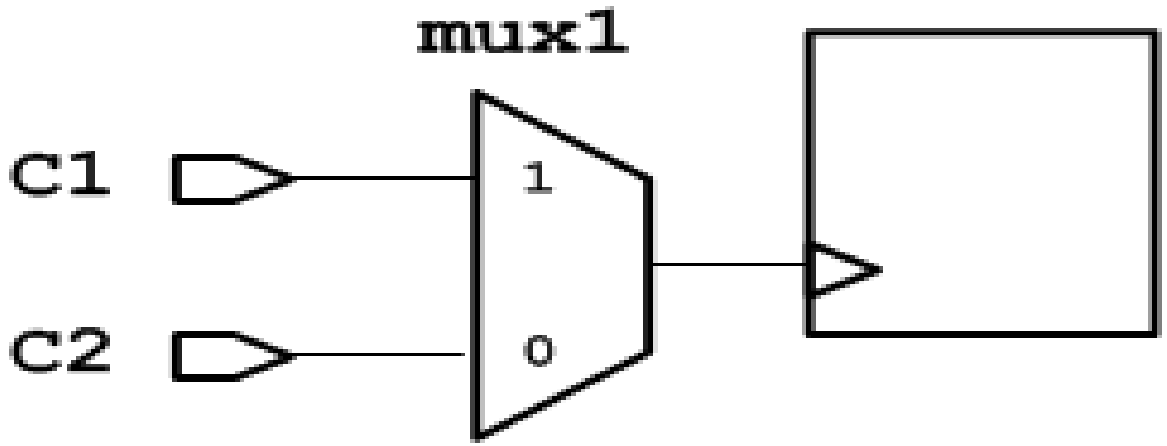


Identify sub-graphs where every node in sub-graph is connected to every other node in sub-graph

SYNOPSYS®

Clock-clock conflict

- Clock in one mode blocks some other clock in another mode



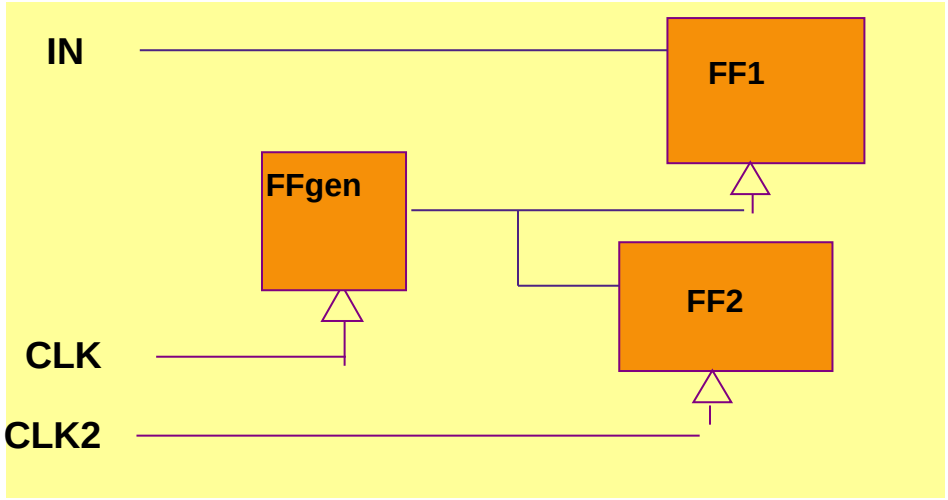

```
# mode1
create_clock –name clk_mode1 [get_ports C1] …
create_generated_clock –name gclk_mode1 \
   [get_pins mux1/Z] …
```

```
# mode2
create_clock –name clk_mode2 [get_ports C1] …
```


SYNOPSYS®

Clock-data conflict

- Clock in one mode blocks data in another mode



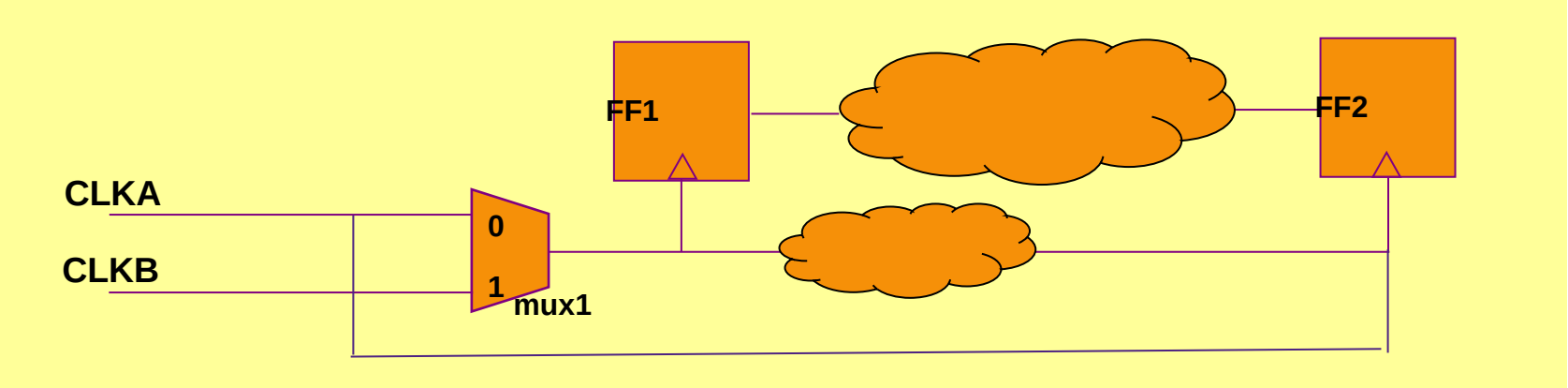

```
# mode1
create_clock –name clk_mode1 [get_ports C1] …
create_generated_clock –name gclk_mode1 \
   [get_pins FFgen/Q] –divide_by 2 …
```

```
# mode2
create_clock –name clk_mode2 [get_ports C1] …
```


SYNOPSYS®

Exception/Path Group Conflict

- Exception in one mode will apply to paths in another mode



Mode 1:
create_clock –name CLKA …
create_clock –name CLKB …
set_multicycle_path 2 –from [FF1/CP]

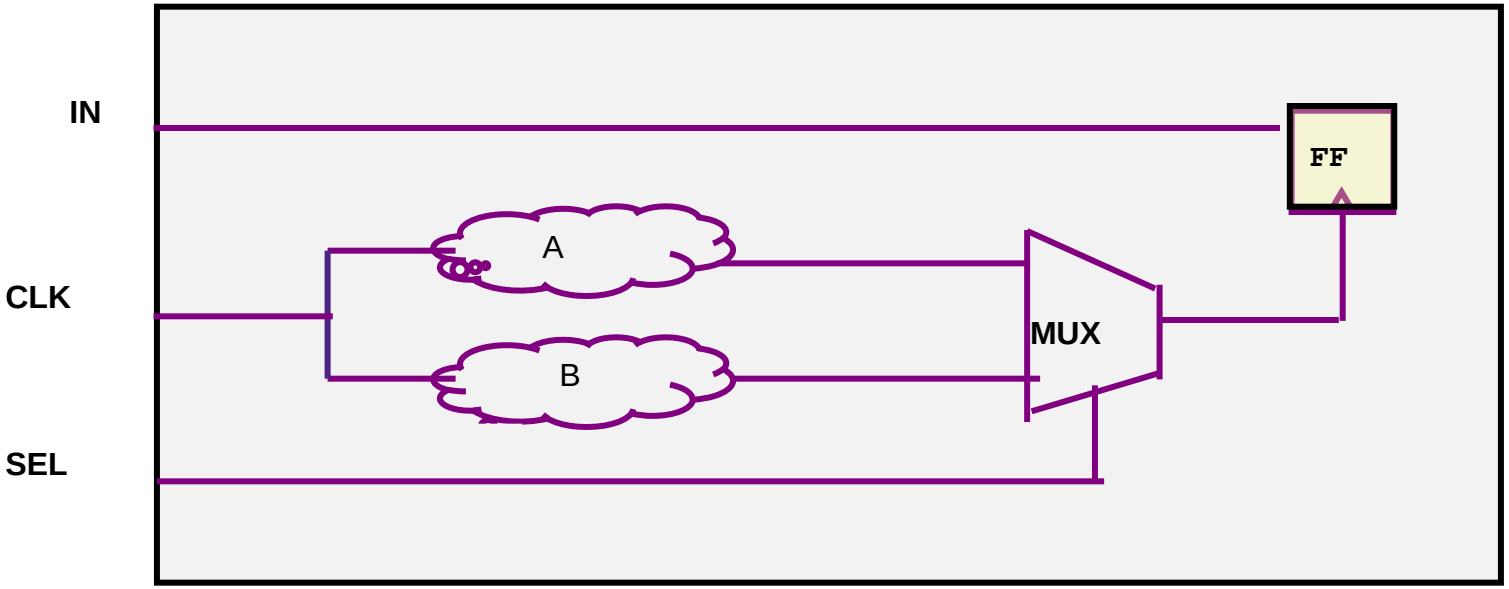
Mode 2:
create_clock –name CLKA …
create_clock –name CLKB …

Mode 1+2:
create_clock –name CLKA …
create_clock –name CLKB …
set_multicycle_path 2 –from [FF1/CP]

SYNOPSYS®

Re-convergence conflict

- Merged mode introduces clock re-convergence not present in individual modes



MODE 1:

create_clock CLK

set_case_analysis 0 [get_port SEL]

MODE 2:

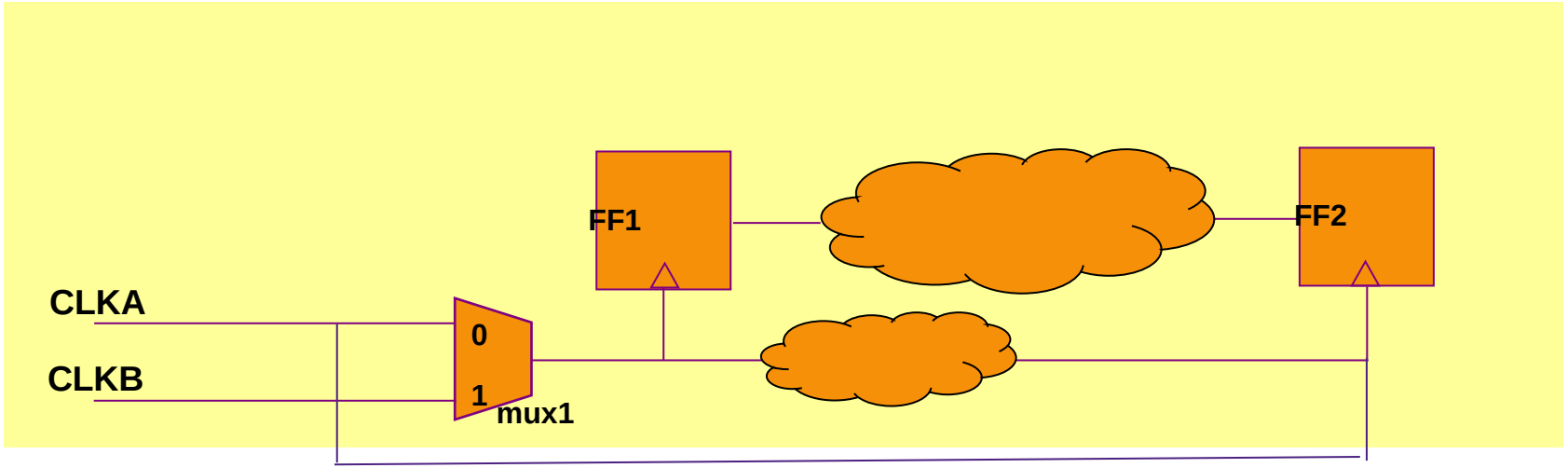
create_clock CLK

set_case_analysis 1 [get_port SEL]

SYNOPSYS®

Value conflict

- Constraint values different in the two modes



Mode 1:
create_clock –name CLKA …
create_clock –name CLKB …
Set_clock_uncertainty 0.5 CLKA

Mode 2:
create_clock –name CLKA …
create_clock –name CLKB …
Set_clock_uncertainty 1.0 CLKA

SYNOPSYS®

Results

Design	Size	Number of Modes		% Reduction	Conflicts found
		Individual	Merged		
A	0.2	95	16	83.1	*clock-clock, exception, value*
B	0.2	3	1	66.6	*None*
C	0.3	12	1	75.0	*None*
D	1.4	3	1	66.6	*None*
E	1.6	5	1	80.0	*None*
F	2.8	3	2	33.3	*clock-clock, exception*
Average				67.5	

Summary

- Mode merging is a key technology to meet the needs of STA for GigaScale, GigaHertz and Giga-Complex designs
- Automatic solution to identify merge able modes is presented

