CMP305
Very Large Scale Integrated Circuit

Lab 2

ASIC Flow (Part 2)

Last Lab

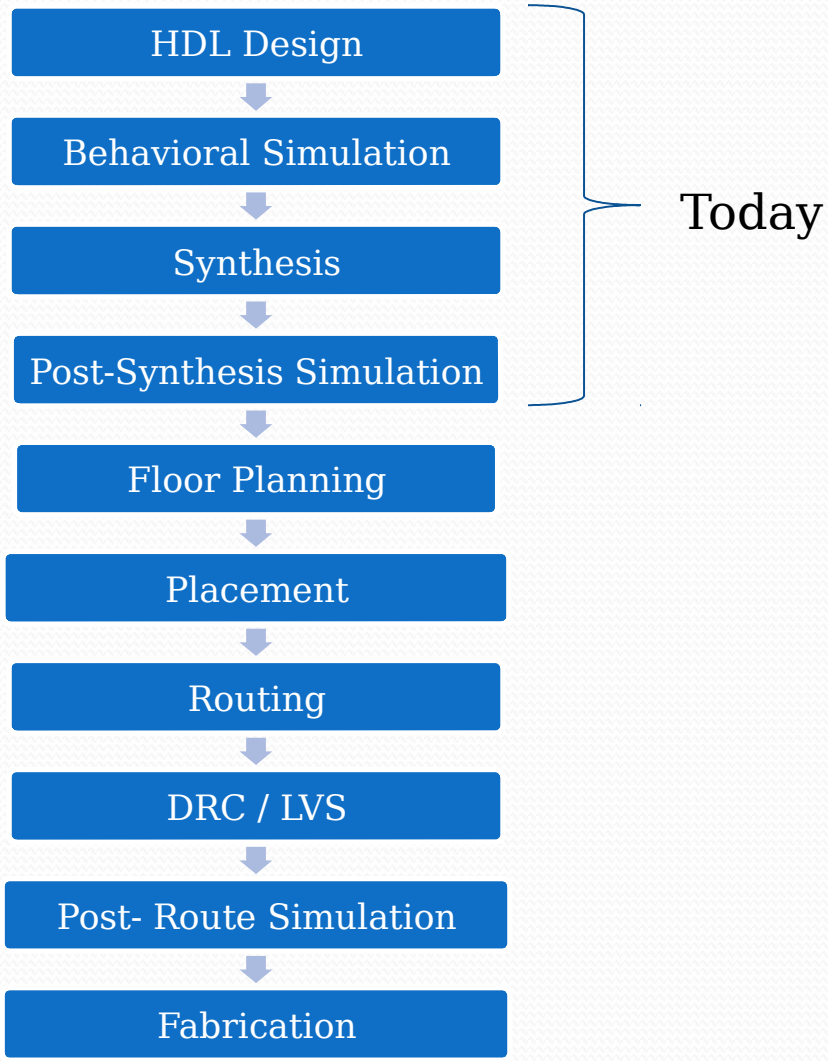
- We Learned about Chip Design Flows
- We Learned about ASIC & FPGA
- We Discussed ASIC Design Flow
- We applied Synthesis Step
- We saw the effect of different designs / Same Functionality

Objectives

- Understand:
  - Static Timing Analysis (STA)
  - Dynamic Timing Analysis (DTA)
  - Design Constraints

- Learn:
  - Use Tcl Scripting
  - Post-Synthesis Simulation

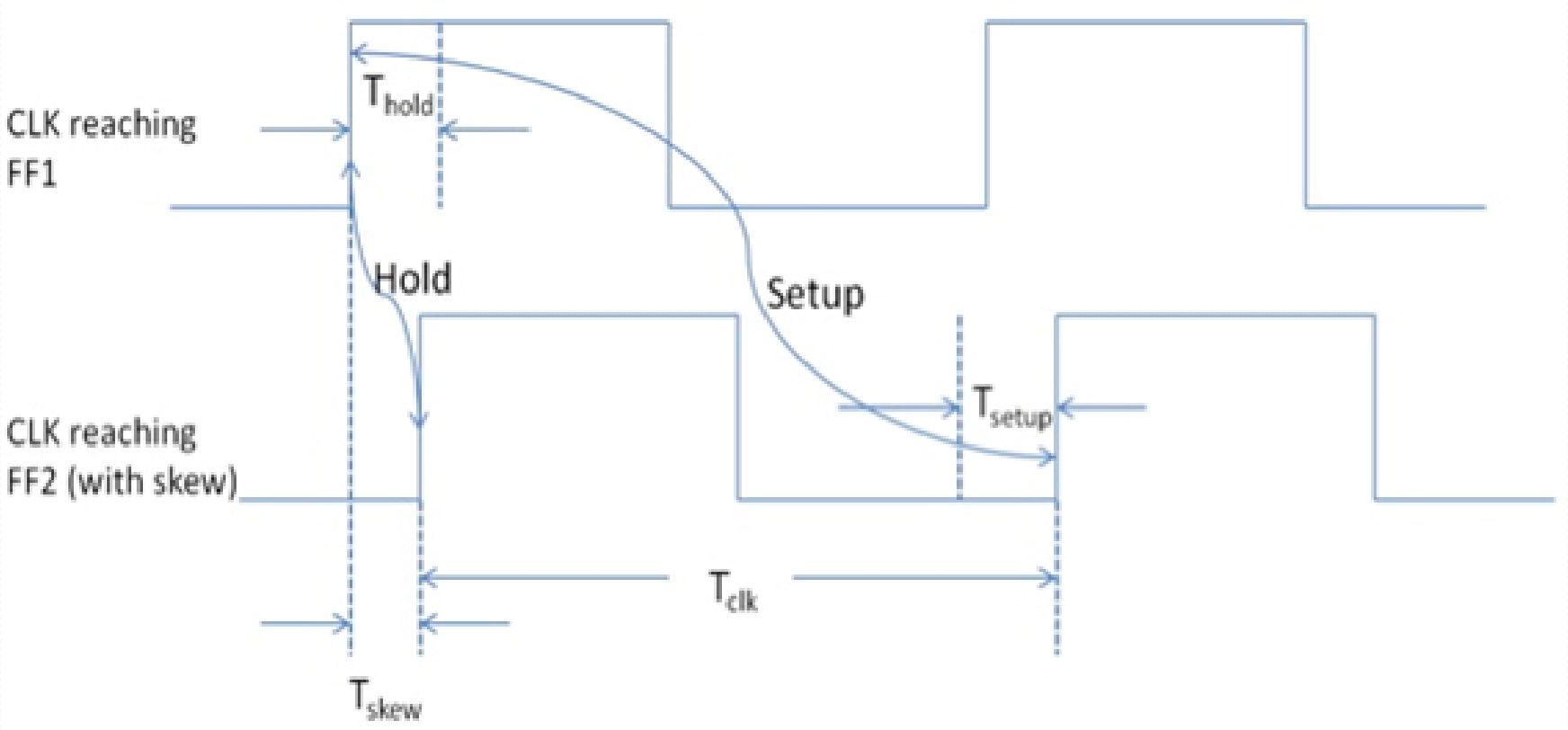
ASIC Design Flow



Timing Analysis

Static Timing Analysis	Dynamic Timing Analysis
Checks static delay requirements	Apply input vectors and check for correct output vectors
Setup / Hold Timing met	Functionality is correct
Run on Synchronous designs only	Synchronous & Asynchronous designs
If wrong, the fabricated IC won’t work properly	If wrong, the design is incorrect
	Must be done before STA
Generated by Leonardo	Simulation on Modelsim

Static Timing Analysis



Static Timing Analysis

- Definitions:
  - Hold Time
    - Min Time the data must be held steady **after** the clock event so that the data is correctly read
  - Setup Time
    - Min Time the data must be held steady **before** the clock event so that the data is correctly read
  - Clock Skew
    - Clock could arrive at different components at different times

Static Timing Analysis

- Definitions:
  - Critical Path
    - The path between an input (Reg) & an output (Reg) with the maximum delay
  - Arrival Time
    - The time instance for a signal to arrive at a certain point
  - Required Time
    - The latest time at which a signal can arrive without making the clock cycle longer than desired
  - Slack
    - The difference between Required & Arrival time.

Static Timing Analysis

- You will study static timing in more details in the lecture

Design Constraints

- System Physics

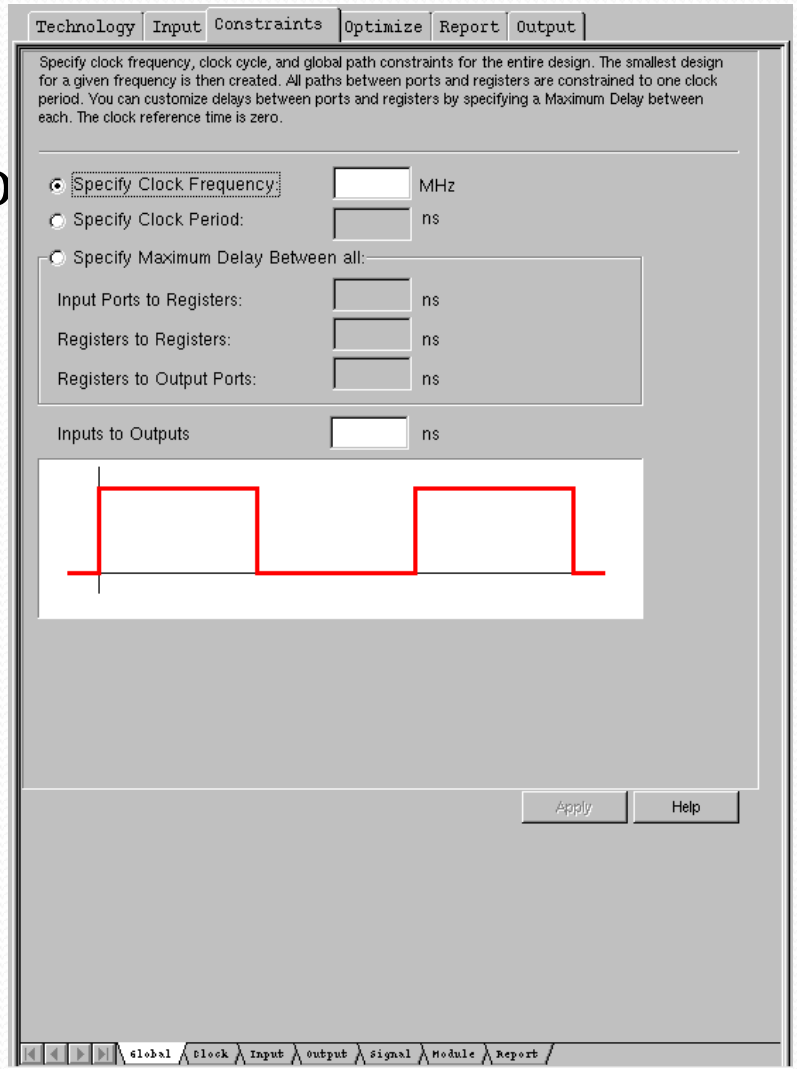
$$P \quad C \cdot V^2 \cdot f \qquad\qquad C \quad \text{Area}$$

Power Capacitance * Voltage^2 * Frequency

- System Requirements
  - Timing Constraints
  - Area Constraints
  - Power Consumption

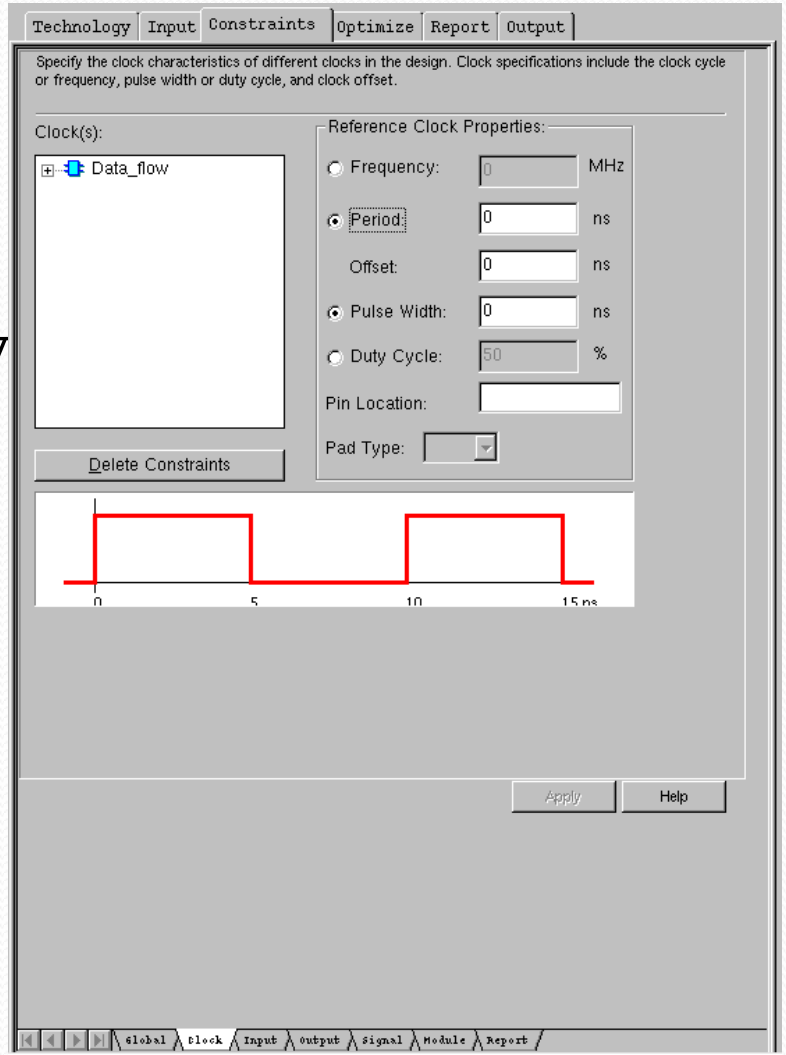
Design Constraints

- General Constraints:
  - Set Clock Freq / Perio
  - Set Max Delay



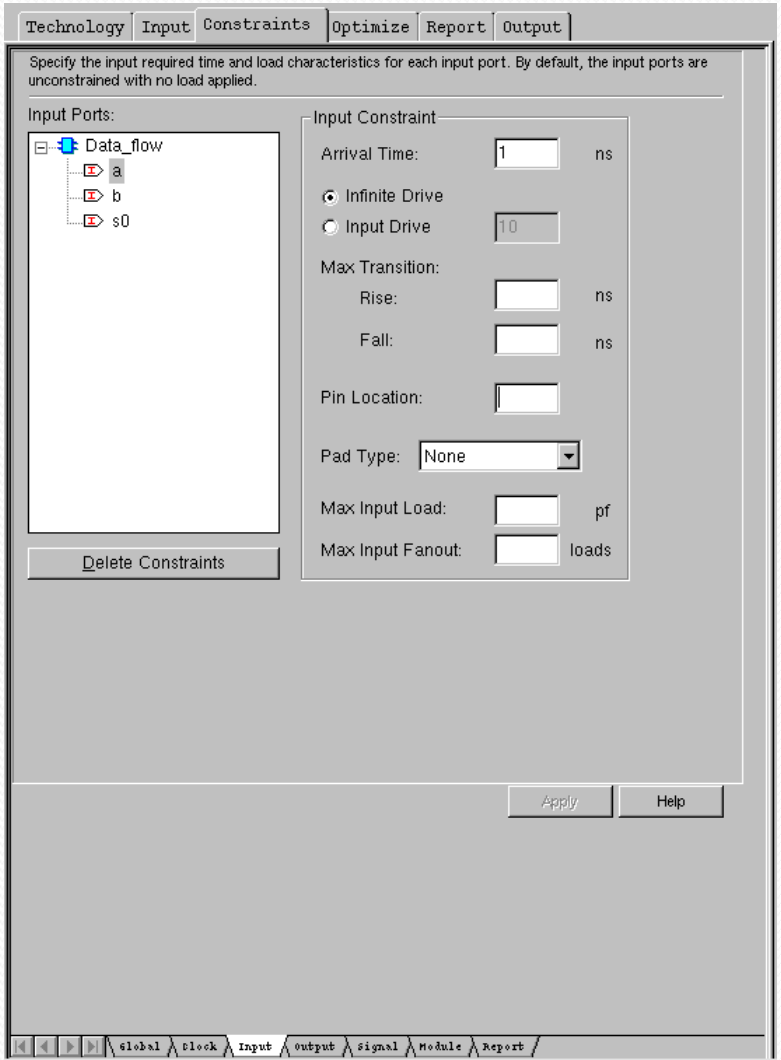
Design Constraints

- Detailed Constraints:
  - Clock Freq / Period
  - Offset
  - Pulse Width / Duty Cy



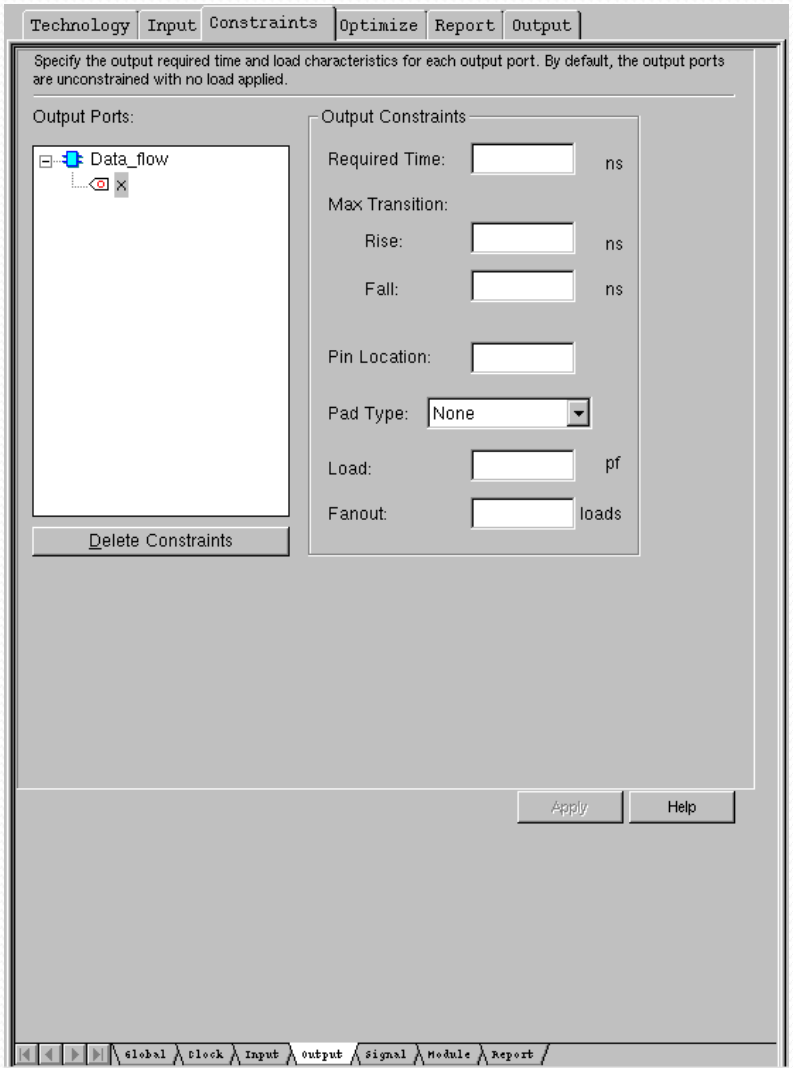
Design Constraints

- Detailed Constraints:
  - Input Arrival Time
  - Max Transition Time
  - Max Fanout
  - Max Load



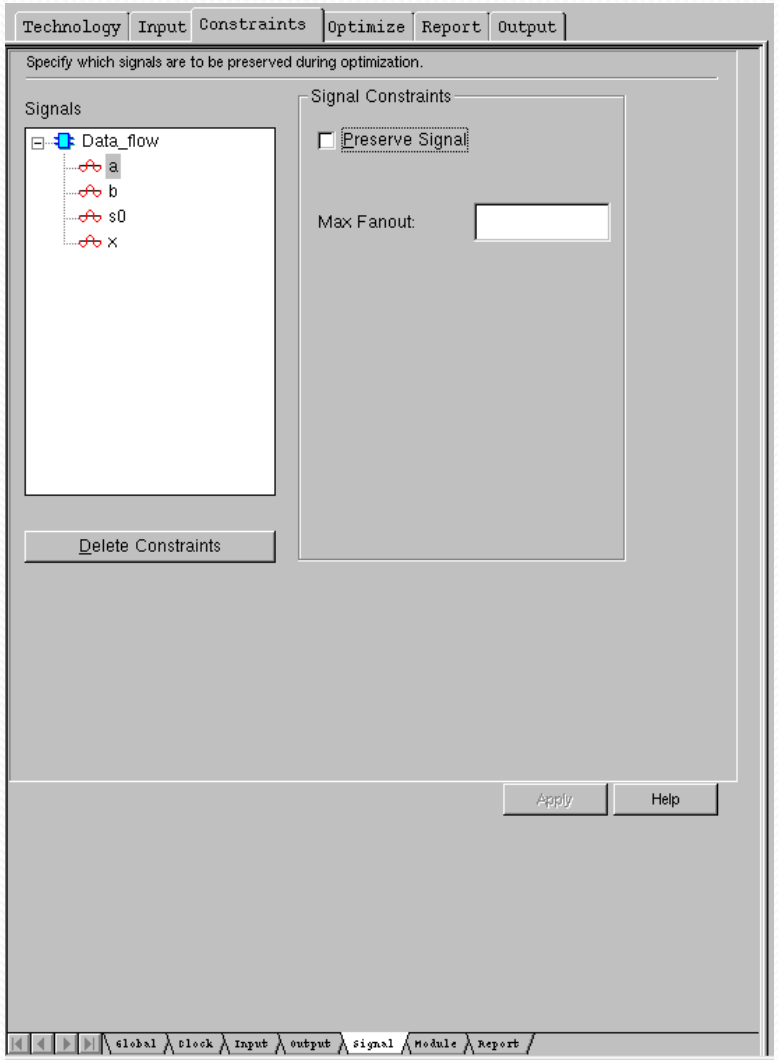
Design Constraints

- Detailed Constraints:
  - Output Required Tim
  - Max Transition Time
  - Estimate Fanout
  - Estimate Load



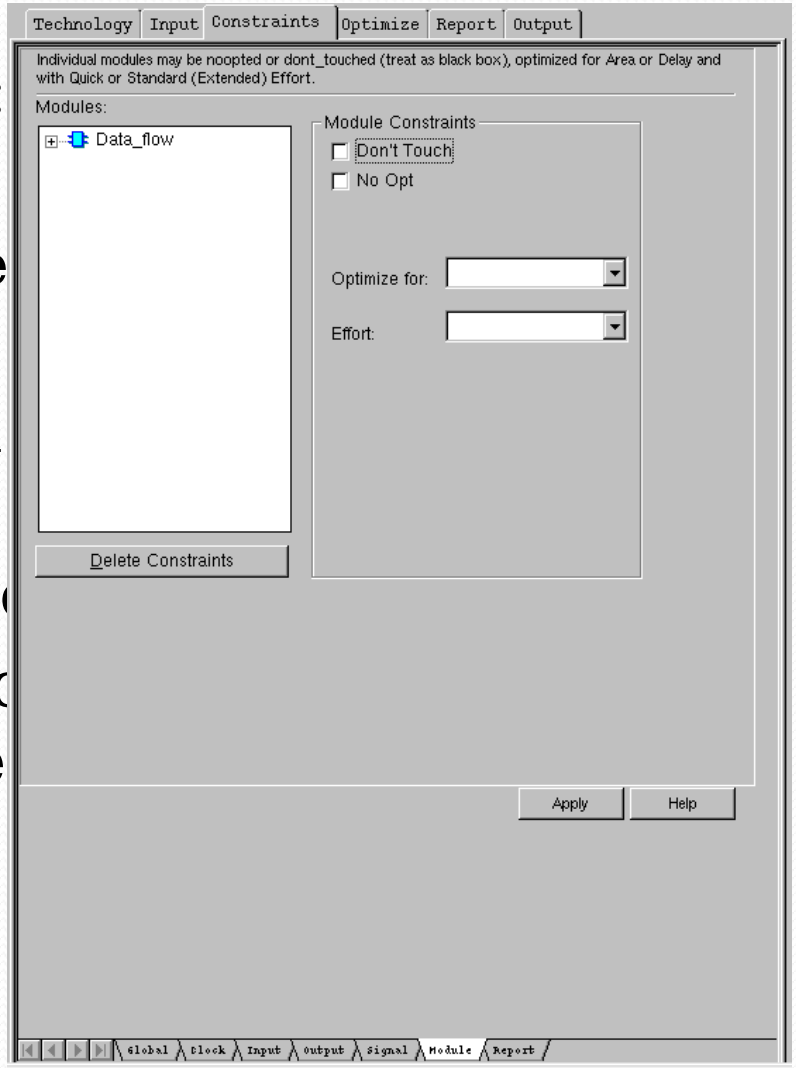
Design Constraints

- Detailed Constraints:
  - Preserve Signal
  - Max Fanout



Design Constraints

- Detailed Constraints:
  - Don’t Touch
    - Will not be synthesize
  - No Opt
    - Will not be Optimized
  - Optimize a specific module for Area/Tim
  - Set Optimization Effo for a specific module



Scripting

- Copy the commands you performed on leonardo
- To run the synthesis script

```
ayman@ubuntu: ~
ayman@ubuntu:~$ spectrum -file Try_mux2.tcl
```

Scripting

1. Load Technology
   - Load_lib tsmc035_typ.syn
   - load_library tsmc035_typ
2. Read Design File
   - read -technology "tsmc035_typ" -dont_elaborate {mux2.vhd}
3. Elaborate
   - elaborate mux2 -architecture Data_flow
   - pre_optimize -common_logic -unused_logic -boundary -xor_comparator_optimize
   - pre_optimize –extract
4. Set Constraint File
   - read_constraints constraint.ctr

Scripting

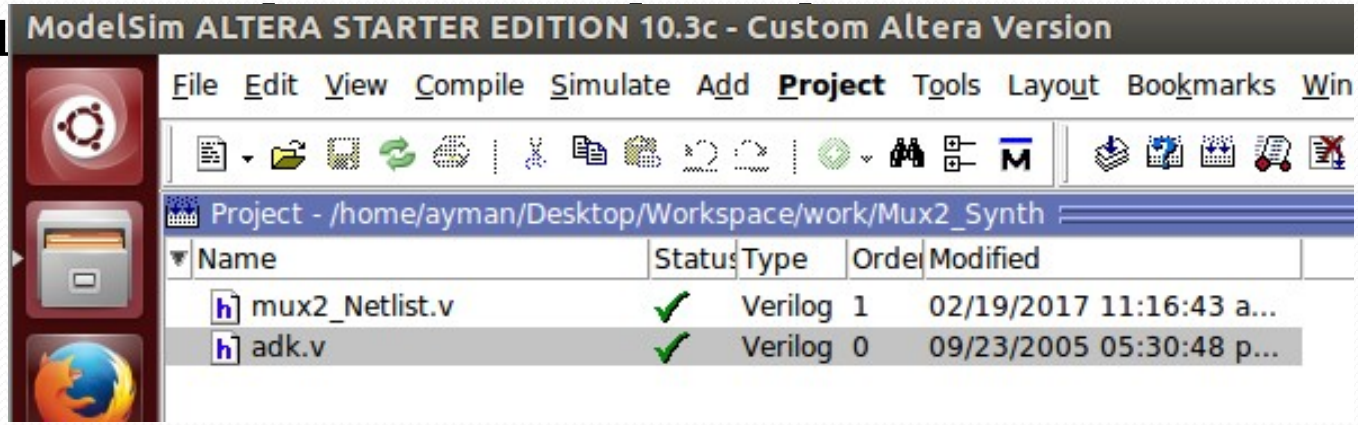
5. Optimize
   - optimize .work.mux2.Data_flow -target tsmc035_typ -macro
     -delay -effort quick -hierarchy auto
6. Optimize Timing
   - optimize_timing .work.mux2.Data_flow
7. Generate Reports
   - report_area Area_Report.rpt -cell_usage -all_leafs
   - report_delay Timing_Report.rpt -num_paths 4 -longest_path -clock_frequency
8. Generate Netlist
   - auto_write -format Verilog ./Outputs/mux2_Netlist.v

Post-Synthesis Simulation

- We want to check if the design is synthesized correctly
  - Un-optimized & badly written Designs could generate wrong functionality
  - Generate Verilog Netlist

Post-Synthesis Simulation

- Create New Project in Modelsim
- Compile the Synthesized design in work library
- Compile Technology File in work library
- Ru



Requirement 1

- Prepare a Constraint File that:
  - Set Max Timing to **5 ns**
  - Set Max Input Arrival Time to **1 ns**
  - Set Max Input Transition to **1 ns**
  - Set Max Input Load to **10 pf**

Requirement 2

- Given n-bit XOR-based Adder (n=8), Use scripts to
  - Synthesis the adder using "TSMC035" Technology
  - Read your constraint file
  - Optimize on Delay & Flatten
  - Report Area, Timing & Netlist
- Is there any Timing Violation? If yes, how to solve it?
- Run Post-Synthesis Simulation

Requirement 3

- Reduce Max Timing to **3 ns**
- Re-run your script
- Is there any Timing Violation? If yes, how to solve it?

Requirement 4

- Reduce Max Timing to **2 ns**
- Re-run your script
- Is there any Timing Violation? If yes, how to solve it?