|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Name | Nguyen Le Thao Vy | No. | 3341 | Div/Dept | DSD/ACD/ACT3 | Job  Date：Dec 15th 2023  Title | Engineer |
| Please tick  the period | First Month | □W1 □W2 □W3 🗹W4 | | | | | |
| Second Month | □W1 □W2 □W3 □W4 | | | | | |
| Third Month | □W1 □W2 □W3 □W4 | | | | | |

1. The weekly report aims to help accelerate member’s workspace integration and should be reviewed by mentor on the last working day of the week.
2. The new weekly report should be reviewed and signed by mentor and direct supervisor.

|  |
| --- |
| Work Experience Record |
| 1. Please describe the tasks and achievements you learned/executed:   All of the tasks I learned during this first week are based on the training plan of 2023, below are a brief description of topics and tasks I was working on:   * Topics of interest:   + Min period / pulsewidth;   + Timescale;   + Simulation - SPICE;   + Testchip workflow. * Tasks that I was working on:   + Read materials – self-learn;   + Work on testchip FSFGFS016A. * Outcomes:   + Learnt basic concepts that aid in working process;   + Got familiar with running design kit and working in project. |
| 1. What are the problems encountered this week? Any actions taken? Any help needed?  * Problem encountered:   + Being unfamiliar with the testchip workflow. * Actions taken:   + Went through the flow in [FTV\_ACD\_TestChip\_Flow\_202303](file:///T:\FTV\DSD\DSD_ACD\ACT4\02_Internal_Tech_Shared\03_Testchip\FTV_ACD_TestChip_Flow_202303) under mentor’s guidance. |
| 1. What are the tasks for next week? Any preparation needed in advance?  * Work on assigned project; * Make time for completing items listed in training plan. |
|  |

|  |  |  |
| --- | --- | --- |
| Name  (Date) | Mentor | Direct Supervisor |
| Nguyen Le Thao Vy  (2023/12/31) | (Signature/Date) | (Signature/Date) |

***Questions from previous review session***

1. Input capacitance of cell.

The so-called “input capacitance” of cell (that is considered in calculating the total capacitive load of an output) refers to parasitic capacitance – which exists between the conducting leads of any component.

Parasitic capacitance: When two electrical conductors are physically close, carry a charge, and there’s a voltage potential between them, they create a *virtual capacitor* between them, even if the conductors are insulated.

1. Explain the impact of poor constraint defining on timing performance? Which phases are affected?

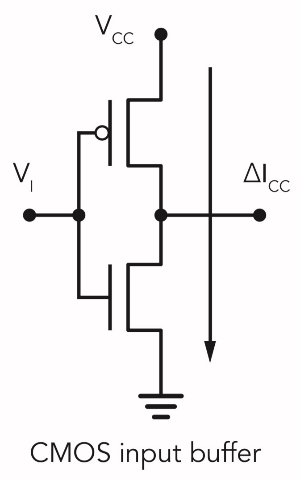
Timing is crucial because it determines how signals propagate through the chip and affects the overall performance and functionality of IC, hence the importance of timing constraints.

Poor constraint defining can be divided into over- and under-constrain. Consider the case of over-constrain, it means that the timing constraints for clock frequency is at least somewhere close to the maximum value permitted, with which the tools would either give up and generate whatever the last pass result, or it may spend an excessive amount of time to generate an optimized version that is significantly slower than using the right constraints, i.e. need 250MHz, ask for 300MHz, get 200MHz. In the case of under-constrain, since the constraints are loosely-defined, potential timing failures can be overlooked and the design may not be optimized correctly / sufficiently.

Phases that can be affected by the quality of timing constraints:

* STA
* Logic synthesis
* Simulation
* Place & route

1. Why floating inputs are more crucial than floating outputs?

Consider a CMOS digital input circuit, when its input signal is either LOW or HIGH, one transistor is on and the other is off, so that it only draws current when it’s switching states.

If V1 is left floating, it will over time have the tendency to accumulate a charge and float toward the logic level change-over point. Once it reaches that point, it causes both the NMOS and PMOS to be partially on, resulting in **excessive current Icc flowing between Vcc and GND**. When the input buffer output switches state, the input then lose charge, causing the circuit to switch back, resulting in the charge hovering around the change-over point and makes the floating input very susceptible to noise, especially from signals switching on adjacent pins.

When is it safe to let an input float? An input can be left floating if the chip / device / cell has an internal pull-up or pull-down resistor, or if designer is purposefully interested in using the floating input.

***Min period – Min pulse width***

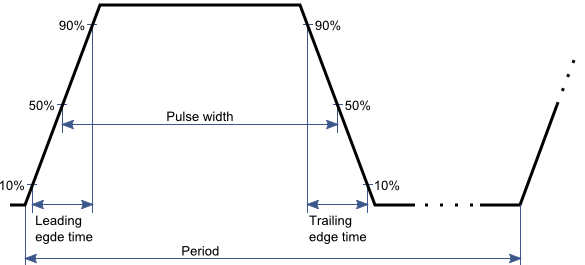
* **Min period check** ensures that the clock driving the circuit does not run at a higher frequency than is tolerated / predefined (the maximum permitted frequency should be found in characteristics datasheet).
* **Min pulse width:** Pulse width can be defined as:
  + In terms of high signal level (high minimum pulse width), it is the time interval between clock signal crossing half the VDD level during rising edge of clock signal and clock signal crossing half the VDD level during falling edge of clock signal.
  + In terms of low signal level (low minimum pulse width), it is the time interval between clock signal crossing half the VDD level during falling edge of the clock signal and clock signal crossing half the VDD level during rising edge of the clock signal.

Figure 1: High pulse width

If the clock being fed to a sequential object has less pulse width than the minimum required a.k.a violates min pulse width, following are the probable outputs:

* + The FF can capture the correct data and FSM will functional correctly;
  + The FF can completely miss the clock pulse and does not capture any new data;
  + The FF can enter metastable state.

Information relates to min pulse width of a cell can be found in liberty (.lib) file or it can be specified in SDC file.

***Timescale***

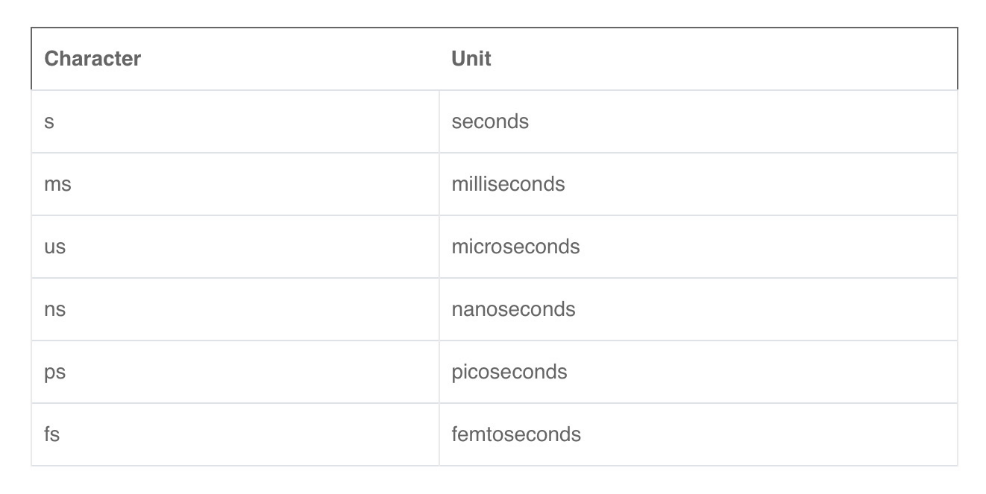
‘timescale <time\_unit>/<time\_precision> is a compiler directive used to specified the time unit and precision for the module that follow it; in which, <time\_unit> is the measurement of delays / simulation time and <time\_precision> refers to how delay values are rounded before being used. Example: ‘timescale 1ns/100ps

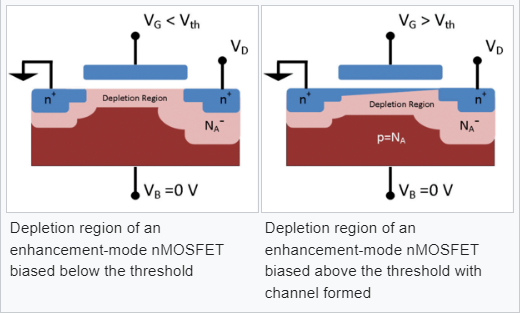
Figure 2: Frequently-used time units

Note: If a delay expression evaluates to an unknown (‘hX) or high-impedance (‘hZ) value, it will be interpreted as zero delay. If one evaluates to a negative value, it will be interpreted as a 2’s complement unsigned integer of the same size as a time variable.

***LVT | RVT | ULVT | SVF | HVT***

* ULVT = Ultra Low Voltage Threshold
* LVT = Lower Voltage Threshold - causes more power consumption and switching timing is optimized; used in time critical functions.
* SVT = Standard Voltage Threshold || RVT = Regular Voltage Threshold - offers trade-off between HVT and LVT i.e. moderate delay and moderate power consumption.
* HVT = Higher Voltage Threshold - causes less power consumption but timing is not optimized; used in power critical functions.

Threshold voltage is the voltage over which, depending on the technology, a certain phenomenon happens. For example, threshold voltage of a MOSFET is the value of the gate voltage when a conductive band forms between the transistor's source and drain.



***Types of simulation - SPICE***

* Device-level simulation – tests the effect of fabrication parameters
* Circuit-level simulation – perform detailed analysis of voltage and current
* Switch-level simulation – treats transistors as switches
* Gate-level simulation – uses gates as the basic components
* Register-transfer-level (RTL) simulation – checks register & combinational logic
* Behavior-level simulation – describes designs in higher level abstraction using hardware description languages or high level languages
* Timing simulation – inspects timing behavior based on a given timing model for logic components and interconnects.
* Fault simulation - checks whether a given set of test vectors could attain a certain level of fault coverage for production test.

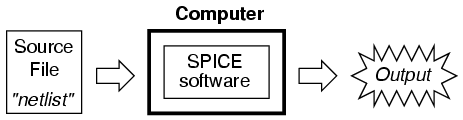
**SPICE** stands for Simulation Program with Integrated Circuit Emphasis. It is a computer simulation and modeling program developed at the University of California at Berkeley, and its main purpose is to mathematically predict the behavioral properties of electronics circuits, with a view to detecting potential issues in design before it is fabricated.

Figure 3: How SPICE works

To run the simulation, a circuit must be presented to SPICE in the form of a netlist. SPICE-based simulators use this netlist to perform circuit calculations, that is, to run the simulation, and outputs data.

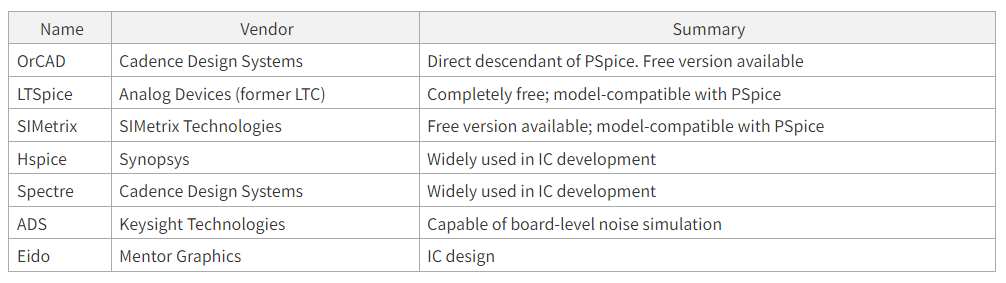
SPICE was developed up through SPICE3 (1985), and thereafter improvements and functions were added to the SPICE foundation for commercial use.

Figure 4: Major simulation software packages

SPICE-based simulators are provided with several functions, such as current / voltage sources, function generators (oscillators), and current / voltage measurement; measurement results can be plotted in graphs…in order to facilitate DC analysis, AC analysis, transient analysis, Monte Carlo analysis, S parameter and Fourier analysis, and noise analysis, among others.