|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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:   + Static Timing Analysis STA;   + Design kit check;   + Testchip workflow. * Tasks that I was working on:   + Study e-courses (3 courses left);   + Do a trial run of testchip flow on existing data;   + Work on testchip FSFGFS016A. * Outcomes:   + Got familiar with running design kit check and working in project. |
| 1. What are the problems encountered this week? Any actions taken? Any help needed?  * Problem encountered:   + Being completely unfamiliar with the testchip workflow;   + Could not run certain tools due to technical limitations. * 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;   + Asked for help in requesting access to other working servers. |
| 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  (2023/12/15) | (Signature/Date) | (Signature/Date) |

***Questions from previous review session***

1. What are the purpose(s) of performing STA?

STA is an approach used to assess / verify the timing performance of the circuit without considering its logical functionality. Performing STA uncovers potential timing violations, ensures that real-life delay shall not affect the functional correctness, and helps in optimizing the performance of the circuit.

1. What is timing arc?

A timing arc represents the timing relationship between pins of logic cells. It is among the components that form a timing path, and can be defined as an indivisible path from one pin to another.

1. What is false path?

False path is a path that exists in the design but is neither functional nor does it need to meet the timing constraints for the design to function correctly. Some examples of false paths are listed below:

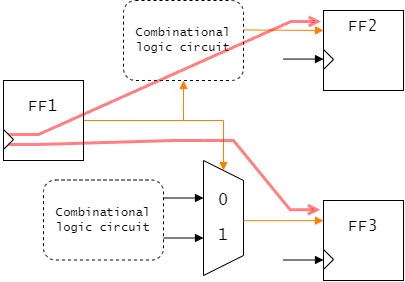
* Paths whose values at startpoint are fixed;

Figure 1: If FF1 remains unchanged thoroughly, then the path from it to other elements are false paths

* Paths in which no data transmission occurs from startpoint to endpoint;

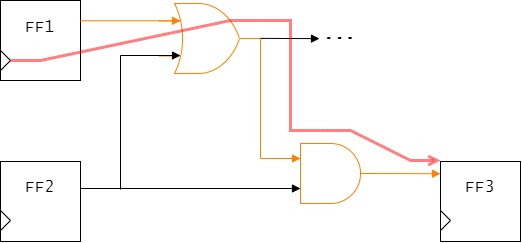


Figure 2: The value from FF1 cannot influence that of FF3 in any scenario

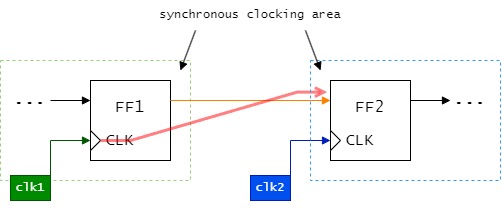
* Paths connecting asynchronous clock domains

Figure 3: Paths between separate synchronous clocking areas

* …

1. What is multicycle path?

Multicycle path is a path that requires (and is permissible to take) more than one clock cycle to complete – which occurs when data being transferred through the path needs to be processed / manipulated before it can be passed on to the next stage.

Difference from false path: Multicycle path is valid and must be analyzed (against more than one clock period).

1. How many types of fault are there?

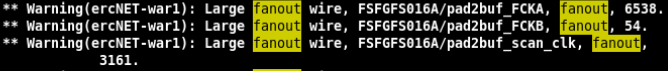
* Transistor faults: Transistor faults can be classified into stuck-open (stuck-off) and stuck-short (stuck-on). As the names suggest, a stuck-open transistor behaves like an open circuit i.e. acting as a resistor, while a stuck-short transistor behaves like a short circuit i.e. operating in conducting mode.
* Stuck-at faults: The value on a faulty signal line appears to be stuck at a constant logic value, either logic 0 (stuck-at-0) or logic 1 (stuck-at-1).
* Delay faults: A delay fault causes excessive delay along a path such that the total propagation delay exceeds the specified limit. There are different delay fault models, namely gate-delay fault, transition fault (this two inspect the time interval taken for a transition from the gate input to its output), path delay fault (considers the cumulative propagation delay along a signal path through the circuit)…
* Crosstalk: Crosstalk is the unwanted coupling (between interconnects) that may result in improper functioning of the circuit.
* …

***Design Kit Check***

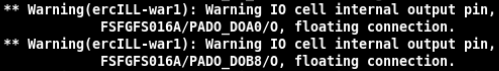
* **FERC –** Electrical Rule Check: FERC is run to verify electrical properties of a design and make sure that it follows a set of electrical rules defined by a designer or industry standards.
  1. Required inputs: Netlist (.v) and design kit setting (setup.ftc)
  2. Outputs: ERC report file (.rep) and log file (ferc.log)
  3. Syntax: *ferc* *-pre*

In working on FSFGFS016A, the below errors and warnings are reported:

* Warning: Large fanout wire 🡪 there exists components in design that exceed the pre-determined MAXFANOUT value in setup file.



* Warning: IO cell internal output pin floating connection 🡪 floating outputs are not critical and are normally left floating.

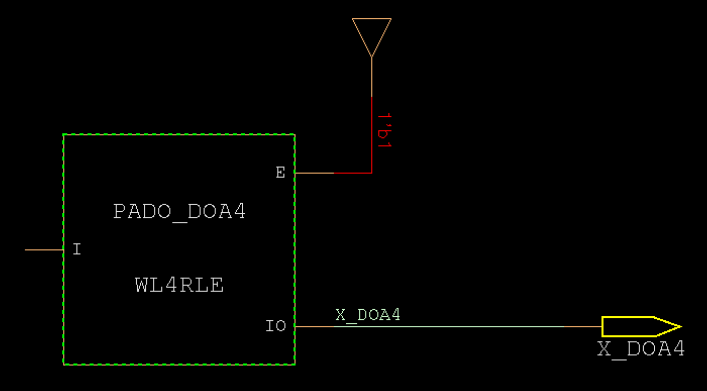


* Warning: Don’t use cell is used in design.



* Error: Floating input / bi-directional pin





* Error: IO cell external pin does not connect to primary inout directly



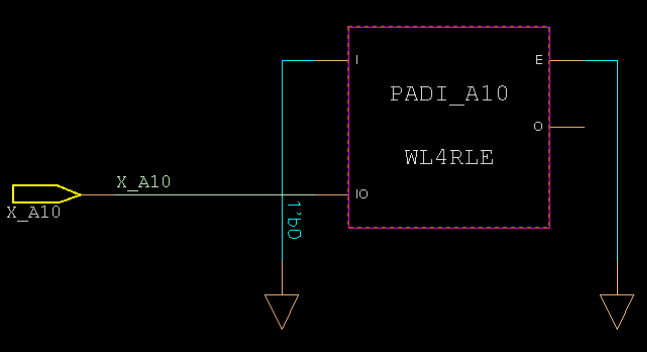
* Error: Primary input / output does not connect to IO cell directly



* Error: Wire together net 🡪 This error occurs when multiple pins are driving one net.

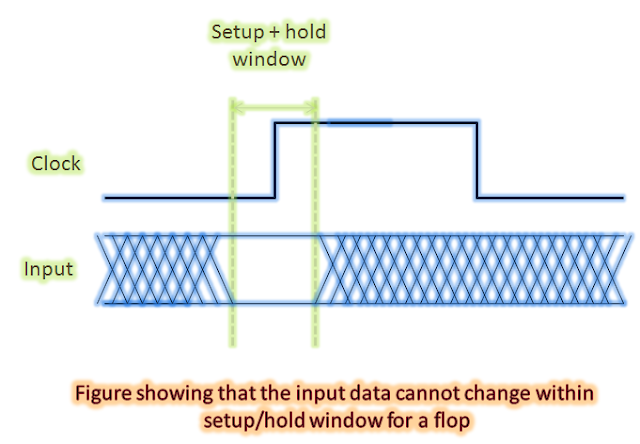


The image attached below shows an input X\_A10 connected to pin IO (defined as inout), that makes the net having 2 input information.



* **FLRE –** Literal Rule Check: flre is run to check names of netlists; the tool will perform literal check / translation, check naming / keyword collisions…to make sure that these names do not violate Faraday’s naming rules.
  1. Required inputs: Netlist (.v) and design kit setting (setup.ftc)
  2. Outputs: a file containing translated Verilog netlist and literal checking results (.v.lre) and log file (flre.log)
  3. Syntax: *flre*
* **FTCV –** Timing Constraint Validation: The purpose of running ftcv is to validate user-provided Synopsys Design Constraint file (.sdc); two stages of this validation include SDC parsing and constraint analysis. In the absence of SDC file, ftcv can still be used to generate a constraint template.
  1. Required inputs: Netlist (.v), SDC file (.sdc), and design kit setting (setup.ftc)
  2. Outputs: Summary files (litmus or ftcv.log.summary) and log file (ftcv.log)
  3. Syntax: *ftcv <mode> <sdc\_id> <analysis\_type> litmus*
* **FSTA –** Static Timing Analysis:
  1. Required inputs: Netlist (.v), SDC file (.sdc), and setup file for fsta (setup.ftc.fsta)
  2. Outputs: Summary file (Constraint.xml) and timing reports (\*rpt)
  3. Syntax: *fstaH –group <sdc\_id> <mode 1..n> <option(s)>*

*fstaH –sta2xml* 🡪 put timing result(s) to .xml file

About the setup and hold time check: Setup and hold timings check are to be met in order to ensure that data launched from one flop is captured properly at the next clock edge.

* Setup: If setup check is violated, data will not be captured at the next clock edge properly, due to data cannot arrive within time period. In the three common timing models (worst / typical / best), setup timing is tested in the worst case, whose setup time is longest.
* Hold: Hold check ensures that the data is held for at least a minimum time, and what is intended to be captured at the next clock edge will not be capture at the same launch edge. Hold timing is tested in the best case.

***MBIST Memory-Built-In-Self-Test***

* **BIST Built-In Self-Test** is a DFT approach that involves inserting additional circuitry – which enables testing without the need for external test equipment – into ICs. The use of BIST helps in lowering the reliance on external ATE, hence reducing testing expense, and is also useful for testing circuits that are not directly connected to external pins.
* **MBIST Memory-Built-In-Self-Test** involves incorporating a self-test circuit within the memory chip to verify memory functionality. The memory cells are then tested by writing a test pattern and reading it back to ensure that it matches the expected result.
* **TestChip MBIST**