**GLS** GATE LEVEL SIMULATION

**FLOW**

Spec🡪architecture🡪RTL design🡪functional verification 🡪

synthesis (LEC-logical equivalent check)🡪DFT🡪STA🡪GLS (without delays or with delays)

**DIIFENTIATE RTL AND GLS**

|  |  |
| --- | --- |
| **RTL** | **GLS** |
| RTL simulation requires RTL code, TB, Tests, Tools | GLS requires netlists, TB, Library files, Tests |
| RTL is higher level abstraction | Lower-level abstraction |
| RTL executes is zero dalay simulation | GLS performs the operation with or without delays |
| Faster simulation | Slower simulation |

**Why is GLS needed?**

* Asynchronous interface
* False Path
* Multicycle path

**We are using both STA and GLS WHY...??**

🡺Using GLS it takes more time to run and debug, so the need will be done in GLS remaining STA.

After GLS, TAPE-OUT.

**TYPES OF GLS**

**NON-TIMING DELAYS (zero and unit delay)**

**ZERO DELAY**

No delays specified initially to get the proper functionality of the rtl provided.

**run time option** 🡺 -nospecify (or) +delay\_mode\_zero

reset and power up sequences.

**ISSUES**

🡺RACE AROUND CONDITIONS

🡺ZERO DELAY LOOP

**UNIT DELAY**

One unit time delay is given to simulate.

**run time option** 🡺 +delay\_mode\_unit

**TIMING GLS**

**SDF SIMULATION**

PD (backend) team will give the file, it includes actual delays, timing checks, timing constraints (like upf)

The delays in sdf file will be annotated into the design.

Actual delays - delays of interconnect cells

Timing checks – hold time, setup time, recovery (sync set up time), removal time (async hold time)

Timing constraints – Various paths, false & multicycle paths

PVT, Incremental, Absolute Delays

**run time option** 🡺 -sdf\_cmd\_line

(or)

$sdf\_annote (“file\_name”, module\_instance)

**Process, Voltage, Temperature - PVT corner**

Details about Chip characteristics

**n-mos - p-mos**

T – T Iopath ay (2 (min),3(typical),4(max))

F – F hold time Iopath ay ((2,3,4) rise (3,4,5) fall (4,5,6,) turnoff)

S – S setup time fall n-mos, rise p-mos

F – S

S - F

Every process has its sdf file.

**ISSUES**

X prop

Set up and hold time

Race around

**WHY RTL TO GLS**

GLS 🡺 X pessimistic

RTL 🡺 X optimistic

**Reasons for X prop**

1. Uninitialized FF/registers
2. Uncommented ports, power supplies (Vdd, Vss)
3. Setup and Hold time Violations.
4. Multiple Drives

**NRF’s (NON RESETABLE FLIP-FLOP)**

Control paths - configuring the registers.

Data paths – to write the data.

Force files

**INPUT TO GLS**

1. Netlist
2. SDF file
3. Force file (2types --- Force deposit and force freeze)

Force deposit – if no value specific initial value will be considered as default. Value will be overridden.

Force freeze --- Fixed value can’t be overridden.

1. Standard cell library

+ve UNEDE -ve UNEDE

Buffer Inverter

And NAND

Or NOR

D ff comb circuit

**TYPES OF DELAYS**

**Transport delay**

Transport delay models propagate all signals to an output after any input signals change. Scheduled output value changes are queued for transport delay models.

**Innertial delay**

Inertial delay models only propagate signals to an output after the input signals have remained unchanged (been stable) for a time period equal to or greater than the propagation delay of the model.

| **STA can miss...** |
| --- |

|  |  |
| --- | --- |
| X-propagation from unknown reset |  |

|  |  |
| --- | --- |
| Glitches or race conditions |  |

|  |
| --- |
| Functional bugs due to synthesis |

|  |
| --- |
| Scan chain issues |

|  |  |
| --- | --- |
| Power-up/reset bugs |  |

| **GLS catches it by...** |
| --- |

Simulating netlist behavior with uninitialized FFs

Timing-based simulation

|  |
| --- |
| Replaying tests on the gate netlist |
| Running scan test patterns with delay |
| Checking real-world startup scenarios |