PROV LOGIC

MASTERING ADVANCED STA: INTERVIEW QUESTIONS WITH EXPERT SOLUTIONS

Prasanthi Chanda

1. What is the impact of OCV (On-Chip Variation) on Setup and Hold Timing Analysis?

OCV Impact on Setup:

- Setup analysis considers the worst-case max delay of the launching path and min delay of the capturing path to account for process variations.
- This ensures that the data arrives within the setup time before the clock edge.

OCV Impact on Hold:

- Hold analysis considers the worst-case min delay of the launching path and max delay of the capturing path.
- This ensures that the data is stable during the hold time after the clock edge.

Mitigation:

- Use Aging Aware STA, Margins (Derates), or Advanced Variability Models like AOCV, POCV, and LVF to handle OCV effects.

2. Why is hold time analyzed at the fastest process corner and setup at the slowest corner?

Setup Time:

- The worst-case for setup occurs when the launching clock path is slowest (slow process, low voltage, high temperature), which increases the clock period.
- The data path must also be slow to avoid violations.

Hold Time:

- The worst-case for hold occurs when the launching clock path is fastest (fast process, high voltage, low temperature), reducing the data arrival time.
- The capturing clock must be slowest to avoid early capture and ensure no hold violations.

3. How do you fix hold violations in a timing path?

Increase Data Path Delay:

- Insert buffers or inverters to increase delay in the data path.
- Use higher VT (threshold voltage) cells to slow down propagation.

Reduce Clock Skew:

- Adjust clock tree synthesis (CTS) to reduce early arrival of the capturing clock.

Reduce Data Path Drive Strength:

- Use a weaker drive strength gate in the data path to slow it down.

Increase Wire Length or Load:

- Add dummy loads (capacitance) or increase wire length for controlled delay.

Modify Metal Layers:

- Adjust metal layers with higher resistance (R) to slow down signals.

4. What is the impact of IR Drop on Timing Closure?

IR drop occurs due to voltage drops in power/ground networks, affecting transistor switching speeds.

Impact on Timing:

- Increased IR drop leads to lower supply voltage (Vdd drop).
- Lower Vdd results in increased cell delay, impacting setup timing.
- If excessive, it can cause timing violations and increase clock uncertainty.

Mitigation Techniques:

- Use power grid optimization, dynamic voltage scaling, and power gating techniques.
- Improve decap placement and route power efficiently in PDN.

5. What is the role of CPPR (Clock Path Pessimism Removal) in Timing Analysis?

CPPR removes excessive pessimism in clock uncertainty when the launch and capture clocks share the same path.

Scenario Without CPPR:

- Clock arrival variations in setup/hold timing add unnecessary pessimism due to derating.

How CPPR Works:

- The common clock path delay between launch and capture paths is subtracted from the uncertainty.
- Ensures realistic analysis, avoiding over-design and unnecessary buffering.

6. What are the implications of Clock Jitter on Timing Closure?

Clock Jitter: Variation in clock edge arrival due to phase noise, power noise, or PLL variations.

Impact on Setup and Hold:

- Setup Violations: If jitter increases, the effective clock period reduces, leading to timing failures.
- Hold Violations: If jitter is excessive, it can alter clock edge alignment, causing incorrect latching.

Mitigation:

- Use low-jitter PLLs, clock buffers with better noise immunity, and minimize power noise to stabilize clocks.

7. How does the Choice of Metal Layers Affect Timing Performance?

Higher Metal Layers (M6, M7+):

- Lower resistance (R), lower delay, suitable for clock and critical data paths.

Lower Metal Layers (M1, M2):

- Higher resistance, used for short local interconnects but adds RC delay if used in long routes.

Optimization Strategy:

- Place high-speed signals on higher metal layers to reduce delay and improve timing margin.
- Reduce signal coupling by using shielding between high-frequency nets.

8. What is CRPR (Clock Reconvergence Pessimism Removal), and when is it required?

CRPR occurs when the same clock path reconverges at different points in the design but is artificially derated multiple times, introducing pessimism.

Example:

- If a clock tree diverges and later reconverges, STA tools may overestimate uncertainty, leading to incorrect timing violations.

Fix:

- Use CRPR-aware STA tools to correct over-pessimistic margins.

9. How does Multi-Corner Multi-Mode (MCMM) Timing Analysis Work?

Multi-Corner Analysis:

- Analyzes timing at different PVT (Process, Voltage, Temperature) corners.
- Ensures design functionality across worst-case conditions.

Multi-Mode Analysis:

- Includes functional modes (normal, low-power, test) to validate constraints under different power and performance settings.

MCMM Strategy:

- Run timing closure across all PVT and functional conditions simultaneously to detect violations early in the design.
- Helps avoid unexpected failures after silicon fabrication

10. What is the difference between Clock Gating and Clock Skew Optimization in STA?

Aspect	Clock Gating	Clock Skew Optimization
Purpose	Power saving	Timing Optimization
Impact	May introduce glitches	Reduces Timing Violations
Methods	Gating cells, enable signals	Buffer insertion, skew balancing

11. How do you handle Crosstalk Induced Delay in Timing Analysis?

Crosstalk Delay Definition:

- Crosstalk occurs when aggressor nets interfere with victim nets, causing delay increase (crosstalk slowdown) or delay reduction (crosstalk speedup).

Impact on Setup and Hold Timing:

- Setup: Crosstalk can increase signal delay, leading to setup violations.
- Hold: Crosstalk can decrease signal delay, leading to hold violations.

Mitigation Strategies:

1. Shielding: Place ground/Vdd lines between aggressor-victim nets.
2. Increase Spacing: Maintain distance between critical nets.
3. Layer Selection: Route timing-critical nets on higher metal layers with less coupling.
4. Buffering: Add buffers to break long victim paths.
5. STA Crosstalk Analysis: Enable SI-aware (Signal Integrity aware) STA tools like PrimeTime-SI.

12. What is the effect of Dynamic IR Drop on Setup and Hold Timing?

Dynamic IR Drop: Caused by sudden switching activity, reducing effective Vdd.

Impact on Timing:

- Setup Timing: Lower Vdd → Slower cell delay → Setup violations.
- Hold Timing: Lower Vdd → Longer hold time requirements → Hold violations.

Fixes:

1. IR Drop-Aware STA: Run IR drop aware timing analysis in signoff tools.
2. Power Grid Strengthening: Increase metal width in power grid design.
3. Decoupling Capacitors (Decaps): Add decaps to smooth power fluctuations.
4. Clock Gating Optimization: Reduce unnecessary switching activity.

13. How Do You Optimize Multi-Cycle Paths (MCP) in STA?

Multi-Cycle Path: Paths where data takes multiple clock cycles to propagate.

Timing Optimization Techniques:

a. Modify Constraints: Set set_multicycle_path in STA to adjust setup/hold checks.

b. Use Pipeline Registers: Break MCPs into multiple register stages to balance delays.

c. Clock Gating Consideration: Ensure MCPs do not get gated unexpectedly.

14. TCL Program to Report the 10 Worst Violating Setup Paths and Save to a File

```
set file [open "setup_violations.txt" w]
puts $file "Worst 10 Setup Violations:"

set paths [report_timing -delay_type max -max_paths 10 -nosplit]
foreach_in_collection path $paths {
    set start_reg [get_attribute $path startpoint]
    set end_reg [get_attribute $path endpoint]
    set slack [get_attribute $path slack]
    puts $file "$start_reg -> $end_reg | Slack: $slack ns"
}
close $file
puts "Report saved to setup_violations.txt"
```

Application: Automates worst setup violation reporting and logs it to a file.

15. TCL Program to Run Timing Analysis for Multiple Corners & Save Reports

```
set corners {"SS" "TT" "FF"}
foreach corner $corners {
    set_analysis_corner $corner
    set file_name "timing_report_$corner.txt"
    set file [open $file_name w]
    puts $file "Timing Report for $corner Corner"
    report_timing -delay_type max -max_paths 5 > $file
    close $file
    puts "Generated $file_name"
}
```

Application: Runs timing for SS, TT, and FF corners and generates separate reports.

16. What are the Challenges of Asynchronous Paths in STA ?

Asynchronous Path Issues:

- No Common Clock Reference: Causes data transfer uncertainty.
- Metastability Risk: Can lead to unpredictable circuit behavior.
- Glitches & Spikes: May result in incorrect data sampling.

Solutions:

- Use Synchronizers: Dual flip-flop synchronizers prevent metastability.

17. TCL Script to Extract Clock Periods and Report Skew

```
set file [open "clock_analysis.txt" w]
puts $file "Clock Name | Period (ns) | Skew (ns) | Latency (ns)"

foreach_in_collection clk [get_clocks *] {
    set period [get_attribute $clk period]
    set skew [get_attribute $clk uncertainty]
    set latency [get_attribute $clk latency]
      puts $file "[get_object_name $clk] | $period | $skew | $latency"
}
close $file
puts "Clock analysis saved to clock_analysis.txt"
```

Application: Extracts clock period, skew, and latency for all clocks.

18. What are AOCV, POCV, and LVF in Advanced Timing Signoff.

- AOCV: Uses fixed derate values per cell type, not path-sensitive.
- POCV: Uses statistical distributions, capturing local and global variations.
- LVF: Provides detailed path-aware delay variations, minimizing over-margining.

19. Python Script to Parse PrimeTime Report and Extract Violations

```
import re

def extract_worst_slack(file_path):
    with open(file_path, "r") as file:
        data = file.readlines()

    violations = []
    for line in data:
                match = re.search(r"slack \(VIOLATED\)\s+(-?\d+\.\d+)", line)
        if match:
            violations.append(float(match.group(1)))

    if violations:
        print(f"Worst Slack: {min(violations)} ns")
    else:
        print("No violations found.")

extract_worst_slack("timing_report.txt")
```

Application: Extracts worst slack from a PrimeTime report.

20. TCL Program to Insert Buffers on Critical Paths with Slack < -0.2ns

```
set slack_threshold -0.2
set file [open "buffer_insertion_report.txt" w]
puts $file "Inserted Buffers on Critical Nets:"

foreach_in_collection net [get_nets -slack_lesser_than $slack_threshold] {
    set net_name [get_object_name $net]
    insert_buffer -cell BUFX2 -net $net
    puts $file "Inserted BUFX2 on $net_name"
}
close $file
puts "Buffer insertion report saved."
```

Application: Optimizes timing by inserting buffers on critical nets.

21. How Does Negative Slack Propagation Impact STA?

When a path has a negative slack, it can impact downstream paths, making closure difficult.

Propagation Scenarios:

- Negative Setup Slack: Causes downstream paths to inherit violations.
- Negative Hold Slack: Makes fixing hold more challenging in the next stage.

22. Python Script to Generate CSV Report of Timing Violations

```
import re
import csv
def extract_violations(file_path, output_csv):
    with open(file_path, "r") as file:
        data = file.readlines()
    violations = []
    for line in data:
        match = re.search(r"Path: (\S+) .* slack \(VIOLATED\)
(-?\d+\.\d+)", line)
        if match:
            violations.append((match.group(1),
float(match.group(2))))
    with open(output_csv, "w", newline="") as csvfile:
        writer = csv.writer(csvfile)
        writer.writerow(["Path", "Slack"])
        writer.writerows(violations)
    print(f"Report saved as {output_csv}")

extract_violations("timing_report.txt",
"timing_violations.csv")
```

Application: Extracts setup and hold violations and saves them in CSV format.

23. TCL Script to Check for Missing Constraints

```
set file [open "missing_constraints.txt" w]
puts $file "Missing Constraints Report:"

if { [llength [get_clocks]] == 0 } {
    puts $file "Error: No clocks defined!"
}
if { [llength [get_timing_exceptions]] == 0 } {
     puts $file "Warning: No false paths or multi-cycle paths defined!"
}
close $file
puts "Missing constraints report saved."
```

Application: Identifies missing constraints like missing clocks or false paths.

24. TCL Script to Report Paths from a Specific Instance

```
set instance "U1"
report_timing -from [get_cells -hierarchical $instance] -max_paths 5 > "timing_report_U1.txt"
puts "Timing report for $instance saved to timing_report_U1.txt"
```

Application: Reports timing violations for a specific instance (e.g., U1).

25. Python Script to Find Most Critical Paths (Worst 5 Violations)

```
import re

def find_critical_paths(file_path):
    with open(file_path, "r") as file:
        data = file.readlines()

    violations = []
    for line in data:
                match = re.search(r"slack \(VIOLATED\)\s+(-?\d+\.\d+)", line)
        if match:
            violations.append(float(match.group(1)))

    violations.sort()
    print("Top 5 Critical Paths:")
    for i in range(min(5, len(violations))):
        print(f"Path {i+1}: Slack {violations[i]} ns")

find_critical_paths("timing_report.txt")
```

Application: Sorts timing violations and finds worst 5 paths.

26. TCL Program to Identify High Fanout Nets

```
set file [open "high_fanout_nets.txt" w]
puts $file "High Fanout Nets (Fanout > 100):"
foreach_in_collection net [get_nets] {
    set fanout [get_attribute $net fanout]
    if { $fanout > 100 } {
        puts $file "[get_object_name $net] | Fanout: $fanout"
    }
}
close $file
puts "High fanout nets saved to high_fanout_nets.txt"
```

Application: Identifies nets with fanout > 100.

27. TCL Script to Generate a Fanout Report for All FF.

```
set file [open "flipflop_fanout.txt" w]
puts $file "Flip-Flop | Fanout Count"
foreach_in_collection reg [get_cells -hierarchical -filter
"is_sequential == true"] {
    set fanout [get_attribute $reg fanout]
    puts $file "[get_object_name $reg] | Fanout: $fanout"
}
close $file
puts "Flip-flop fanout report saved."
```

28. TCL Script to Identify the Worst Hold Violations and Save to a File

```
set file [open "hold_violations.txt" w]
puts $file "Worst 10 Hold Violations:"
set paths [report_timing -delay_type min -max_paths 10 -nosplit]
foreach_in_collection path $paths {
    set start_reg [get_attribute $path startpoint]
    set end_reg [get_attribute $path endpoint]
    set slack [get_attribute $path slack]
    puts $file "$start_reg -> $end_reg | Slack: $slack ns"
}
close $file
puts "Report saved to hold_violations.txt"
```

Application: Extracts and saves worst 10 hold violations.

29. What is CRPR (Clock Reconvergence Pessimism Removal)?

CRPR corrects excessive pessimism in timing calculations caused by common clock paths in launch and capture clocks. It removes unnecessary extra delays from timing analysis.

30. TCL Script to Detect Unconstrained Paths

```
set file [open "unconstrained_paths.txt" w]
puts $file "Unconstrained Paths:"
foreach_in_collection path [get_timing_paths -
no_constraints] {
    set start_reg [get_attribute $path startpoint]
    set end_reg [get_attribute $path endpoint]
    puts $file "$start_reg -> $end_reg is unconstrained!"
}
close $file
puts "Report saved to unconstrained_paths.txt"
```

Application: Identifies paths without timing constraints.

31. How does OCV (On-Chip Variation) impact STA?

OCV accounts for process variations, causing data and clock paths to behave differently. It introduces setup pessimism (slower data, faster clock) and hold optimism (faster data, slower clock), requiring derating factors.

32. What is the impact of a high transition time on STA?

High transition time (slow signal rise/fall) increases delay and noise susceptibility, violating timing and signal integrity constraints.

33. Python Script to Parse and Extract Clock Domains from SDC File

```
import re

def extract_clocks(sdc_file):
    with open(sdc_file, "r") as file:
        data = file.readlines()

    clocks = []
    for line in data:
        match = re.search(r"create_clock -name (\S+) -period
(\d+\.\d+)", line)
        if match:
            clocks.append((match.group(1),
float(match.group(2))))

    for clk in clocks:
        print(f"Clock: {clk[0]}, Period: {clk[1]} ns")

extract_clocks("constraints.sdc")
```

Application: Extracts clock names and their periods from an SDC file.

34. Python Script to Compare Two Timing Reports and Find Differences

```
def compare_reports(file1, file2):
    with open(file1, "r") as f1, open(file2, "r") as f2:
        report1 = set(f1.readlines())
        report2 = set(f2.readlines())

    added = report2 - report1
    removed = report1 - report2

    print("New Violations Added:")
    for line in added:
        print(line.strip())

    print("\nViolations Removed:")
    for line in removed:
        print(line.strip())

compare_reports("timing_report_before.txt",
"timing_report_after.txt")
```

Application: Compares before and after optimization reports.

35. TCL Script to Identify & Report Large Net Delays

```
set file [open "large_net_delays.txt" w]
puts $file "Nets with Delay > 2 ns:"

foreach_in_collection net [get_nets] {
    set delay [get_attribute $net delay]
    if { $delay > 2.0 } {
        puts $file "[get_object_name $net] | Delay: $delay ns"
    }
}

close $file
puts "Large net delays report saved."
```

Application: Finds nets with delay > 2ns.

36. What is the difference between Global and Local Skew?

- Global Skew: Clock arrival difference across entire design (e.g., between two flip-flops in different blocks).
- Local Skew: Clock arrival difference between flip-flops within the same clock domain.

37. Python Script to Extract Paths with Negative Slack from a Report

```
import re

def extract_negative_slack(file_path):
    with open(file_path, "r") as file:
        data = file.readlines()

    violations = []
    for line in data:
        match = re.search(r"slack \(VIOLATED\)\s+(-\d+\.\d+)",
line)
        if match:
            violations.append(float(match.group(1)))

    print(f"Number of violations: {len(violations)}")
    print(f"Worst slack: {min(violations)} ns")

extract_negative_slack("timing_report.txt")
```

Application: Filters only negative slack violations.

38. Why is Hold Fixing Done After Setup Fixing?

Setup fixing involves reducing path delays, which can worsen hold time. Hold fixing adds buffers to short paths, ensuring hold violations are corrected without affecting setup.

39. What is a Timing Arc?

A timing arc defines how delay propagates inside a cell (e.g., input to output delays in standard cells). It is modeled in Liberty (.lib) files.

40. What is the purpose of Multicycle Paths in STA?

Multicycle paths allow data to be captured after multiple clock cycles, reducing setup constraints and enabling longer delay paths without violations.

41. What is the difference between Ideal Clock and Propagated Clock in STA?

- Ideal Clock: No propagation delay is considered; arrival time is manually defined.
- Propagated Clock: Actual clock network delays are considered during timing analysis.

42. What are False Paths in STA?

False paths are non-functional timing paths that don't affect circuit operation. They are ignored in STA using set_false_path.

43. What is the difference between Latch-Based and Flip-Flop-Based STA?

- Flip-Flop STA: Analyzed using edge-triggered constraints (setup/hold).
- Latch-Based STA: Uses level-sensitive analysis, making it more complex due to timing window dependencies.

44. How does clock uncertainty impact high-speed processor design?

Clock uncertainty (caused by jitter, skew, and margin variations) affects timing closure in high-speed processors. To mitigate its impact, techniques like clock gating, skew balancing, and on-chip PLL compensation are used. In multi-GHz processors, even a 10ps jitter can cause timing violations, leading to setup failures in critical paths.

45. Why is OCV (On-Chip Variation) more critical in 7nm and below technologies?

At sub-7nm nodes, manufacturing variations (PVT: Process, Voltage, Temperature) become dominant, causing timing fluctuations. This leads to increased setup and hold margin mismatches. Adaptive voltage scaling (AVS) and dynamic OCV compensation (LVF models) are used to counter these variations dynamically in modern chip designs.

46. Explain how multi-corner, multi-mode (MCMM) analysis ensures reliable chip operation.

MCMM ensures a chip functions correctly under all scenarios (fast/slow silicon, high/low temperature, different functional modes). Without MCMM, a chip optimized only for typical conditions might fail in worst-case corners (e.g., low voltage + high temperature).

Example: A mobile processor operates in turbo mode (high frequency, high power) and power-saving mode (low frequency, low power). MCMM analysis ensures both modes meet timing.

47. How does Metal Stack Selection Affect Timing Closure in ASIC Design?

Different metal layers have different RC parasitics. High-speed signals should use upper metal layers (M8, M9) for lower resistance, while power and ground should use wider metal layers for efficient distribution. Incorrect metal selection can lead to higher delay, impacting setup slack and causing hold timing failures.

48. How do you optimize a design with negative slack in deep submicron technology?

- Fix Setup Violations: Reduce data path delay using buffer insertion, gate sizing, or restructuring logic.
- Fix Hold Violations: Insert delay cells or increase clock skew at violating paths.
- Use DFT Techniques: Scan chains and pipeline registers reduce timing stress.
- Post-route Optimization: ECO (Engineering Change Order) fixes specific timing failures post-layout.

Application: In AI accelerators, negative slack paths degrade inference speed, so designers apply aggressive hold fixing to improve stability.

49. What is the significance of pulse width checks in clock trees?

- Pulse width checks ensure the clock high/low phases are long enough to avoid setup and hold violations in flip-flops. If a pulse width is too short, the FF may not latch data correctly, leading to functional errors.
- Example: In a DDR memory controller, insufficient pulse width causes data corruption at high speeds (>3200 MT/s).

50. Why is clock domain crossing (CDC) verification crucial in automotive chips?

- CDC errors in automotive SoCs (e.g., ADAS processors) can cause random failures due to metastability.
- Example: A self-driving car's camera module communicates at 200 MHz, while its AI processor runs at 1 GHz. Improper CDC handling could cause image processing glitches, leading to incorrect object detection.

Solutions:

- FIFO Synchronization: Buffers data when crossing between domains.
- Handshake Mechanisms: Ensures controlled data transfer.
- Dual-Flip-Flop Synchronization: Reduces metastability risk.

Excellence in World class VLSI Training & Placements

Do follow for updates & enquires




+91- 9182280927