Scan-Chain Analysis and Tempus Tcl Command Document

1. **Define the Clock**

```

create_clock -name clk -period 5 [get_ports clk]

```

2. Scan Chains in the Netlist

(i) How many scan chains?
There are two primary scan chains:

- **Chain 1:** from f32_data_reg_0_ to f32_data_reg_10_ (scan out: test_so1).

```

# For Chain 1 scan input from port to first flip-flop

report_timing -from i_agg_su_count_a[31] -to f32_data_reg_0_

# For Chain 1 scan output from the last flip-flop to the output port

report_timing -to test_so1 -unconstrained

```

- **Chain 2:** from f32_data_reg_11_ through many cells (ending at o_spare_config_reg_31_) with additional intermediate “isolation” taps (scan_iso_or, scan_iso_or1, …, scan_iso_or9).
  In total, Chain2 provides ten additional tap outputs.

```

# For Chain 2 scan input

report_timing -from test_si1
```

For Chain 2 final scan output

report_timing -to o_spare_config[31] -unconstrained

Isolation tapouts reports

report_timing -to scan_iso_or -unconstrained

report_timing -to scan_iso_or1 -unconstrained

report_timing -to scan_iso_or2 -unconstrained

report_timing -to scan_iso_or3 -unconstrained

report_timing -to scan_iso_or4 -unconstrained

report_timing -to scan_iso_or5 -unconstrained

report_timing -to scan_iso_or6 -unconstrained

report_timing -to scan_iso_or7 -unconstrained

report_timing -to scan_iso_or8 -unconstrained

report_timing -to scan_iso_or9 -unconstrained

```

**(ii) List of scan chain input ports:**

**I_agg_su_count_a[31]**

**test_si1**

- For Chain1: The scan data is shifted in from the bit **i_agg_su_count_a[31]**.

- For Chain2: The scan input is determined by a multiplexer that selects between **f32_data_reg_12_.Q**, **test_si1**, using select line **raw_scan_en**.
```

(iii) List of scan chain output ports:

test_so1

o_spare_config[31]

- Chain1's serial output is **test_so1**.
- Chain2's final output is **o_spare_config[31]**.
- Additionally, the chain is tapped at intermediate points to provide outputs: **scan_iso_or, scan_iso_or1, scan_iso_or2, scan_iso_or3, scan_iso_or4, scan_iso_or5, scan_iso_or6, scan_iso_or7, scan_iso_or8,** and **scan_iso_or9**.

3. Reset Path (for a flip-flop with reset)

(iv) Trace the reset pin:
Take for example the flip-flop instance **f32_mux_1_data_reg_29_**. Its reset pin (RN) is driven by a network that originates from the primary reset input **i_reset_**. In the netlist, i_reset_ is buffered and inverted (using BUFX8, INVX2, and INVX4 cells) ,which then drive the RN pin of scan-enabled flip-flop.
During scan mode, i_reset_ is held inactive (logic high for an active-low reset) so that the reset does not clear the scan chain.

Commands

```

report_timing -from i_reset_ -to f32_mux_1_data_reg_29_/RN

```

Path :

Instance	Arc	Cell	Delay	Arrival Time	Required Time
-	i_reset_ ^	-	-	0.000	4.750
U13	A ^ -> Y v	INVX4	0.049	0.049	4.800
U14	A v -> Y ^	INVX2	0.076	0.125	4.876
U17	A ^ -> Y ^	BUFX8	0.169	0.294	5.045
f32_mux_1_data_reg_29_	RN ^	SDFFRX1	0.000	0.294	5.045