



Technical University of Munich
School of Computation, Information and Technology
Chair of Electronic Design Automation

Buffering strategy and floorplan optimization for feedthroughs in a partition

Research Internship Report

Karan Kedia, Guenther Schroeder



Technical University of Munich
School of Computation, Information and Technology
Chair of Electronic Design Automation

Buffering strategy and floorplan optimization for feedthroughs in a partition

Research Internship Report

Karan Kedia, Guenther Schroeder

Advisor : Meng Lian
Advising Professor : Prof. Dr.-Ing. Ulf Schlichtmann
Topic issued : 22.04.2024
Working period : 22.04.2024 - 15.7.2024

Karan Kedia, Guenther Schroeder
Arcisstraße 21
80333 München

Abstract

In Very Large Scale Integration (VLSI) Physical Design, it is a common to have feedthrough bus through a partition that carry data to the next partition but also to the partition itself. These occur in tile like architectures. These structures, often suffer from congestion problems, timing problems, routing resource constraints etc. This project aims to develop ways to tackle these problems at floor planning and feedthrough stages.

Contents

1. Introduction and Background	6
1.1. Introduction	6
1.2. Background	6
1.2.1. Drive strength	7
1.2.2. Why inverters instead of buffer cells?	7
2. Floorplanning	8
2.1. Re-connections needed for the modified floorplans	9
2.2. Interleaved memory-buffer stages floorplan	11
2.3. Semi-interleaved memory-buffer stages floorplan	12
2.4. Everything to the side	13
3. Feedthrough strategies	15
3.1. Normalized bus timing results	17
3.1.1. Scenario 1 - Low voltage and slew, room temperature, Cmax corner	17
3.1.2. Scenario 2 - Low voltage and slew, cold temperature, crosstalk corner	18
3.1.3. Scenario 3 - Low voltage and slew, cold temperature, Cmax corner	19
3.1.4. Scenario 4 - High voltage and slew, cold temperature, crosstalk corner	19
3.1.5. Scenario 5 - High voltage, low slew, cold temperature, Cmax corner	20
4. Results and Observations	22
4.1. Non-timing results	22
4.2. Timing results	23
4.2.1. Why are the WNS and TNS numbers bad?	24
4.3. Future Scope	24
Bibliography	25

List of Figures

1.1. Feedthrough bus	6
1.2. Pipelined Feedthrough bus	7
2.1. Baseline Floorplan	9
2.2. Re-connections after even number of stages	10
2.3. Re-connections after odd number of stages	10
2.4. Interleaved memory-buffer stages floorplan	11
2.5. Semi-interleaved memory-buffer stages floorplan	12
2.6. Memory on the side of the buffer stages	13
3.1. Four stages of DS4 buffers	16
3.2. 2 stages of DS2 and 2 stages of DS4	16
3.3. 2 stages of DS1 and 2 stages of DS4	16
3.4. Boxplot of low voltage and slew rate, room temperature, capacitive worst scenario	17
3.5. Boxplot of low voltage and slew rate, cold temperature, crosstalk worst corner - scenario	18
3.6. Boxplot of low voltage and slew rate, cold temperature, capacitive worst scenario	19
3.7. Boxplot of High voltage and slew rate, cold temperature, crosstalk worst corner scenario	20
3.8. Boxplot of High voltage, low slew rate, cold temperature, capacitive worst corner scenario	21

List of Tables

4.1. Non-timing results	22
4.2. Timing Results	23

1. Introduction and Background

1.1. Introduction

This project focuses on optimizing floorplan and buffering strategies for partitions that contain feedthroughs that carry data and clock signals to the partition and successive partitions, alias feedthrough bus. Figure 1.1 illustrates the feedthrough bus.

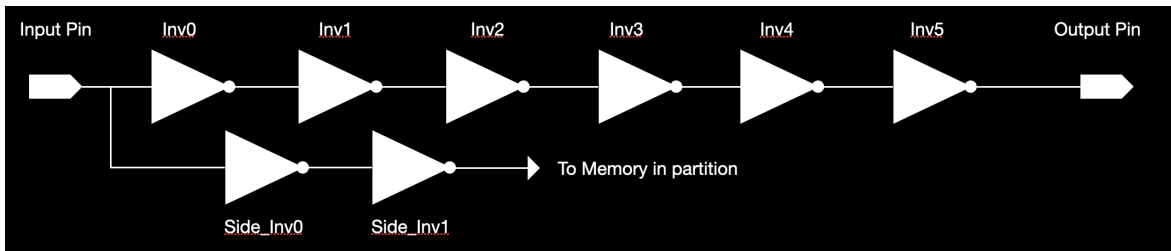


Figure 1.1.: Feedthrough bus

Multiple different floorplan and feedthrough strategies were explored during this project, and normalized results are presented in this report.

1.2. Background

Feedthroughs are wires/nets that are pushed down to a partition from the top chip level. These nets don't interact with the logic within the partition but need to be buffered depending on the Manhattan distance between the input and output pins of the feedthrough in the partition. [1] In some cases (commonly tiling architectures) feedthroughs can carry the data through the partition but the same data is also supplied to the partition itself. In such cases, the feedthrough bus can also be pipelined to avoid large delays as shown in Figure 1.2

These implementations often have timing problems or congestion and routing resource problems depending on the width of the feedthrough bus. Choosing the right buffering strategy can also be a challenge. The decision between using more stages of lower drive strength buffers versus less stages of higher drive strength buffers is also a challenge.

1. Introduction and Background

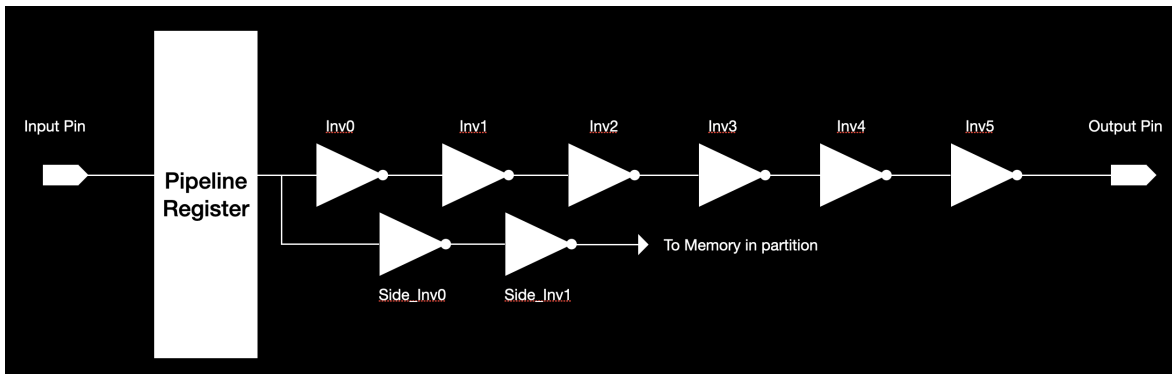


Figure 1.2.: Pipelined Feedthrough bus

1.2.1. Drive strength

Drive strength refers to how much current a standard cell can push at its output when it switches on.[7] For example, in finFET technology, the number of fins in the standard cell is usually associated with the drive strength. This project, used buffers of 4 different drive strengths. DS1 being the lowest drive strength and DS4 being the highest.

1.2.2. Why inverters instead of buffer cells?

The inverter stages in the Figures 1.1 and 1.2 are referred to as buffer stages throughout the report. This is because even though these are inverters, they are used as buffer stages. There are a few benefits to this:-

- A buffer is essentially two successive inverters which means that buffers would have more delay than an inverter. This would mean that if buffer cells were used instead of inverters, there would be more delay on the same path. Since the same number of buffer stages would be needed.
- Using buffers would also change the data duty cycle uncontrollably, while inverters would provide comparatively more control over the data duty cycle.
- Using inverters also saves power, since one buffer would also consume roughly 2 inverters worth of power.

Doing this is possible here because this is a feedthrough and the data on these lines isn't used elsewhere in the partition. This replacement of inverters for buffers would not work in most other cases since it could lead to data correctness issues or use up more space than needed.

2. Floorplanning

Floorplanning is the most important stage of the Physical Design flow. A good floorplan can prevent congestion and hotspot problems during and after placement, and routability and routing resource problems during routing. [4] For example, if the macros are placed in the middle of the design, one might observe routability problems because macros often have routing blockages for several metal layers above the macro. Similarly if there are islands of empty space surrounded by macros, the placement tool might not place cells in that area leading to wastage of area or if it does place cells there, it might experience routability issues. Consequently, a bad floorplan can lead to many other issues like area wastage, high power and heat dissipation, routing congestion, clock distribution, etc.

The Baseline floorplan from which this project started is shown in Figure 2.1. The Figure shows the approximate location of the components involved in this project. The blocks in the figure represent the components as follows-

- The purple bar is the input pins.
- The green bar is the output pins.
- The blue box is the pipeline registers shown in Figure 1.2.
- The red block is the memory banks that receive the data from the inputs.
- The yellow blocks are the regions where buffers for each stage are placed. This is not a macro but the region in which the buffers for each stage are placed.

There are other components in the partition that are not indicated in the Figure since these are irrelevant to this project.

2. Floorplanning

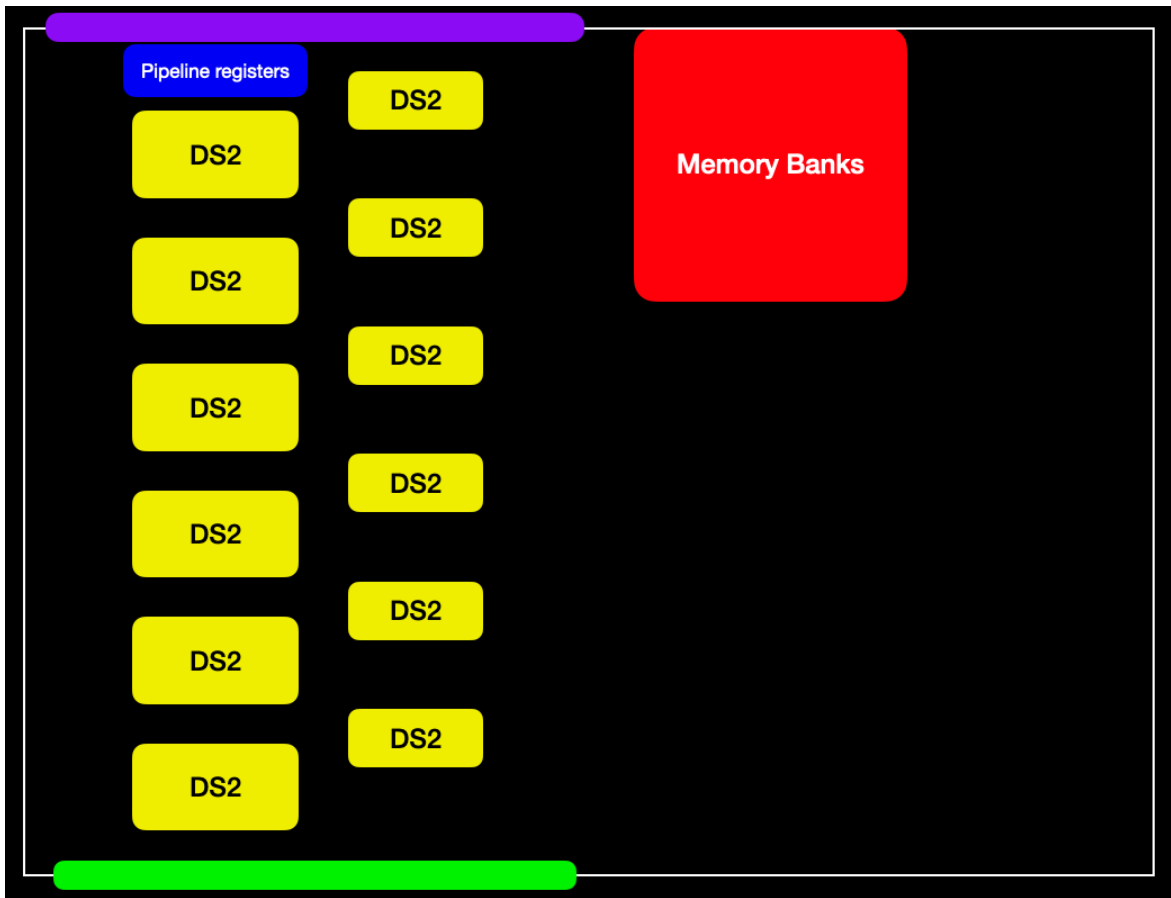


Figure 2.1.: Baseline Floorplan

2.1. Re-connections needed for the modified floorplans

Before the subsequent floorplans were constructed, certain re-connections were made to prevent routing resource, and timing problems.

In Figure 1.2 the side inverters diverge from the feedthrough immediately after the pipeline registers. This is not ideal for routing resources in the region immediately below the pipeline registers (see Figure. 2.1). This can also necessitate reserving a big area near the pipeline registers, and in the middle of the partition for memory, like in Figure. 2.1.

To have more flexibility in floorplanning, the side inverters were reconnected between the buffer stages. The re-connections made are shown in Figures 2.2 and 2.3. Figure 2.2 shows the case where the side inverters are connected after an even inverter stage. Since it would be the correct signal after an even inverter stage, both the side inverters are kept. Figure 2.3 shows the case where the side inverters are connected after an odd inverter stage. Since after an off inverter stage, the signal would be inverted. Only one side inverter is kept, to invert the signal again.

2. Floorplanning

Here, to prevent the risk of the placer placing the side inverters in a 'convenient' location and interfering with the feedthrough line, the side inverters were pre-placed immediately next to their Fan-in inverter. This would shorten the wire from the input inverter to the side inverter and not add to the load significantly.

These re-connections were crucial for developing the 3 floorplans described in the next sections.

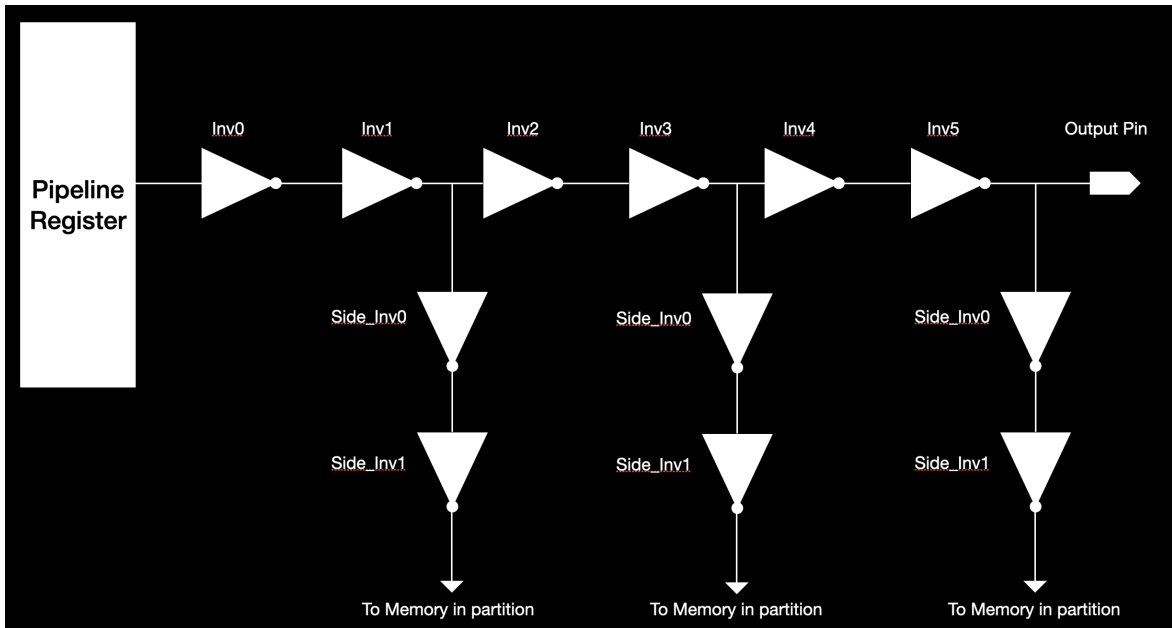


Figure 2.2.: Re-connections after even number of stages

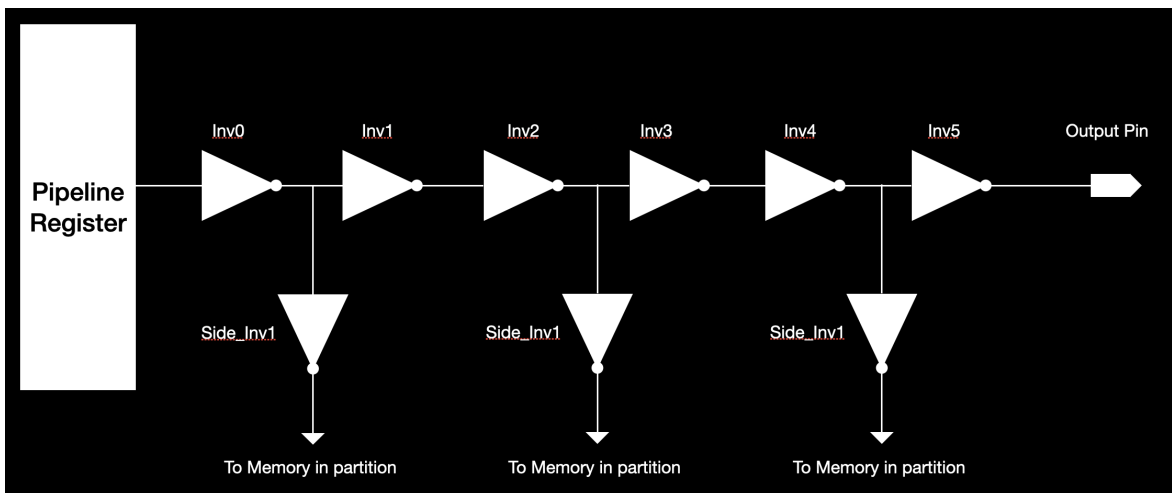


Figure 2.3.: Re-connections after odd number of stages

2.2. Interleaved memory-buffer stages floorplan

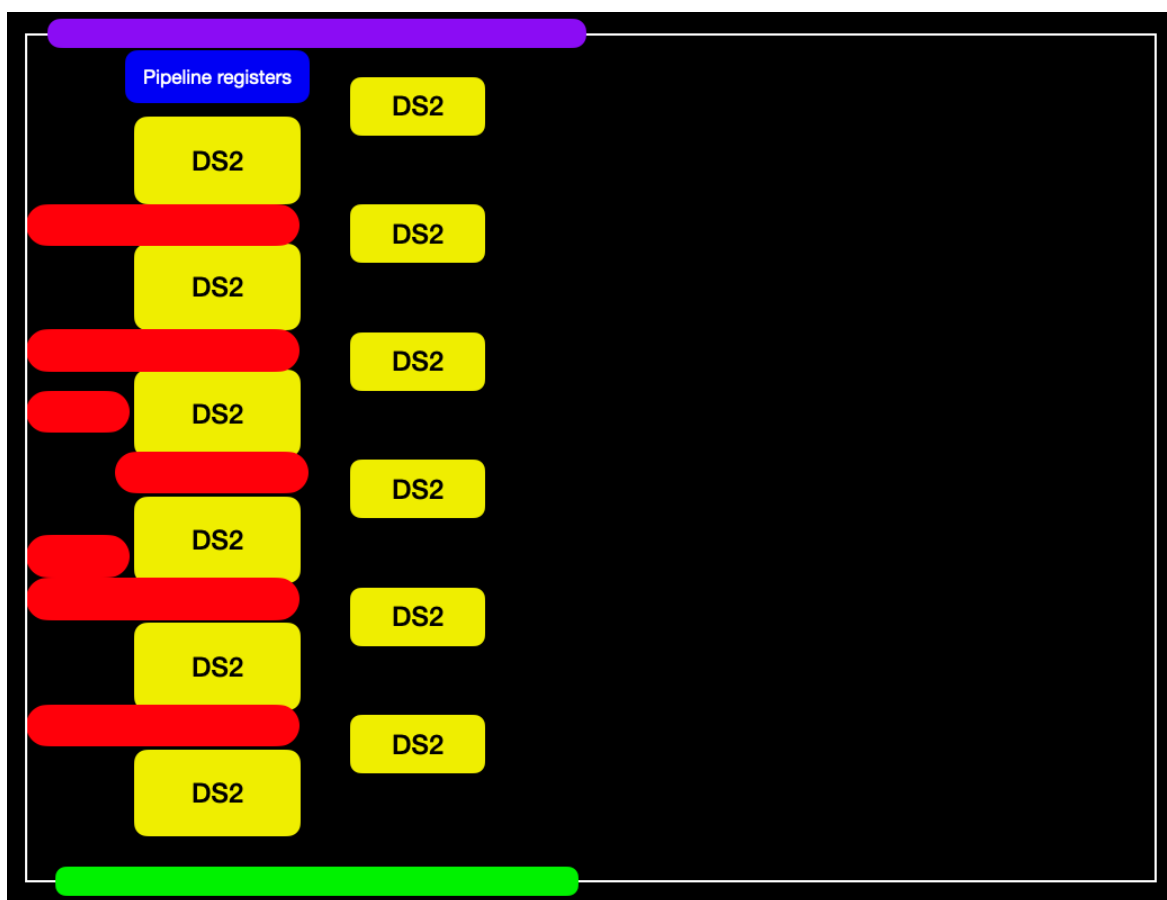


Figure 2.4.: Interleaved memory-buffer stages floorplan

With the flexibility provided by the re-connections made in the previous section, the first floorplan aimed at reducing the distance between the memory and their respective side inverters. Figure 2.4 shows how this was achieved. The memory was arranged in rows and placed in between the buffer stages. This brought the memories as close to their respective side inverters as possible. The floorplan was named FP_blockbreaker since the rows of memory look like the blockbreaker game.

The advantages of this floorplan are listed as follows:

- The spaces between the buffer stages are routing resource constrained spaces. Placing the memory there prevents standard cells from being placed in a resource constrained space.
- The memory has routing blockages up to a certain metal layer, this would force the feedthrough wires to pass over the memory, using thicker wires with less parasitics.

2. Floorplanning

The floorplan also comes with some pitfalls as listed below:

- This creates islands wherein there is space for the placer to place cells, and route them, but which is surrounded by space that does not have enough routing resources, like the space to the left of the yellow boxes.
- The floorplan is now dependent on the buffering strategy used.

2.3. Semi-interleaved memory-buffer stages floorplan



Figure 2.5.: Semi-interleaved memory-buffer stages floorplan

Some of the problems with the previous floorplan were tried to fix with this floorplan. The memory is now arranged in regions that were islands in the previous floorplan, while still being slightly interleaved with the buffer stages. This floorplan was named FP_ttris since here, the memories look like the ttris blocks.

The benefits of doing this are:

2. Floorplanning

- Retains the advantages from the previous floorplan to a certain extent.
- Covers off the islands created by the previous floorplan.

The downsides of this floorplan are:

- The placer is now free to place standard cells in the resource constrained space present between the buffer stages.
- The floorplan is still dependent on the buffering strategy.

2.4. Everything to the side

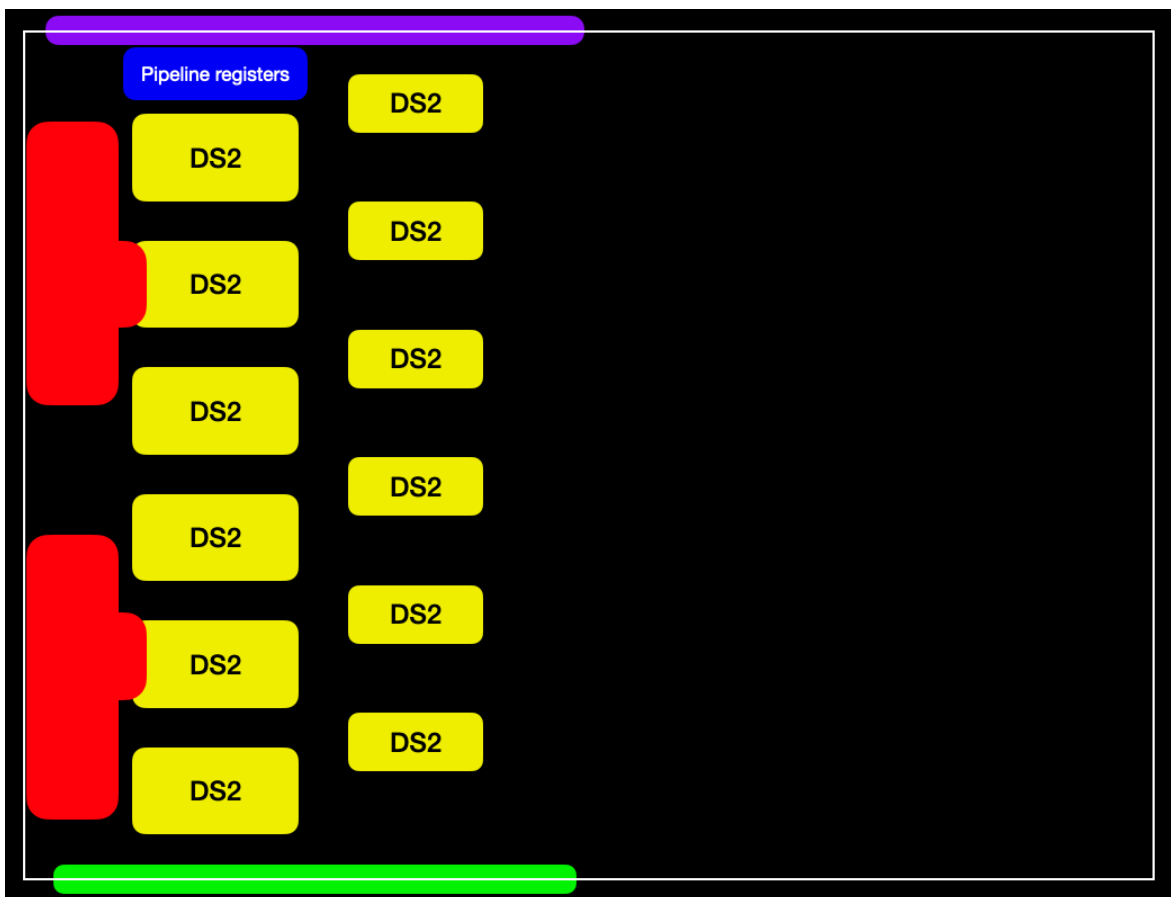


Figure 2.6.: Memory on the side of the buffer stages

This floorplan is aimed at eliminating the dependence of the floorplan on the buffer stages. The interleaving of the memory with the buffer stages made the floorplan dependent on the buffering strategy. The goal of interleaving the memory with the buffer stages was to reduce

2. Floorplanning

the distance between the memories and the side channel inverters. But with the re-connections done in Figures. 2.2 and 2.3 the memory will still be close to the inputs without the interleaving. Hence, the interleaving was removed and all the memory was moved to the left side of the buffer stages, as shown in Fig. 2.6. This floorplan is called FP_ttris_v2.

Advantages of doing this are-

- This gives the placement tool a big uninterrupted rectangular area for the remainder of the standard cells.
- Makes the floorplan modular which enables experimentation on different floorplan strategies.

Disadvantages of this floorplan are-

- Created two islands above and below the memories which are still isolated from the rest of the partition and the standard cells in that area can suffer from routing congestions.
- Loses all benefits of the interleaved floorplan.

3. Feedthrough strategies

With the memories out of the way of the buffer stages, as depicted by the floorplan in Figure. 2.6, experiments were conducted on the buffer stages of the feedthrough. There is no pre-established relation between the drive strength of a standard cell and how far (physically) the next cell can be before we start to see diminishing returns. To find an approximate value for this distance, the load values for an acceptable delay were picked and divided by the resistive and capacitive load per unit distance of the wire. This gives an approximate value (closer to the maximum) for the distance that a specific driver can drive. Through this process it was found that the driver with DS1 can drive a distance of 1 (normalized distances), DS2 can drive about 1.9 to 2.1, DS3 can drive about 2.4 to 2.7 and DS4 can drive roughly a distance of 3. Using this information, several configurations were tried for the feedthrough part, as listed below:

- 6xDS2 (baseline)
- 4xDS4 - Figure.3.1
- 4xDS3
- 2xDS2 2xDS3
- 2xDS2 2xDS4 - Figure.3.2
- 2xDS3 2xDS4
- 2xDS1 2xDS4 - Figure.3.3
- 2xDS2 2xDS1 2xDS2
- 4xDS1 2xDS3
- 4xDS1 2xDS4
- 2xDS1 2xDS2 2xDS3

Here, since the different drive strength buffers can drive different distances, the distance between the buffer stages was adjusted according to the drive strength of the buffer chosen. An example of this can be seen in Figures 3.1, 3.2 and 3.3

3. Feedthrough strategies

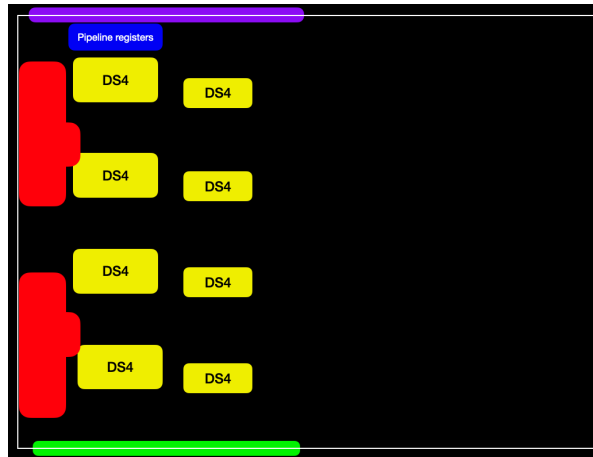


Figure 3.1.: Four stages of DS4 buffers

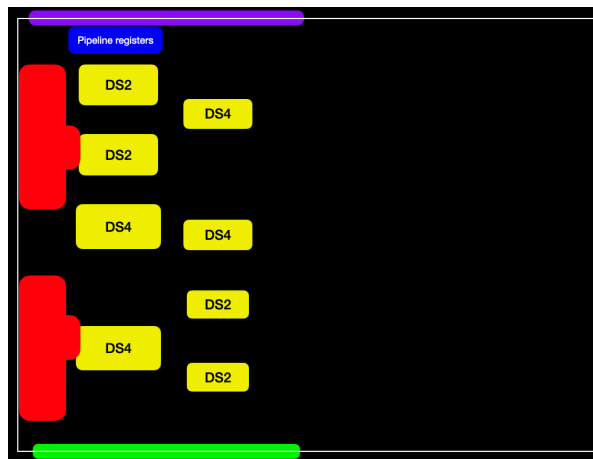


Figure 3.2.: 2 stages of DS2 and 2 stages of DS4

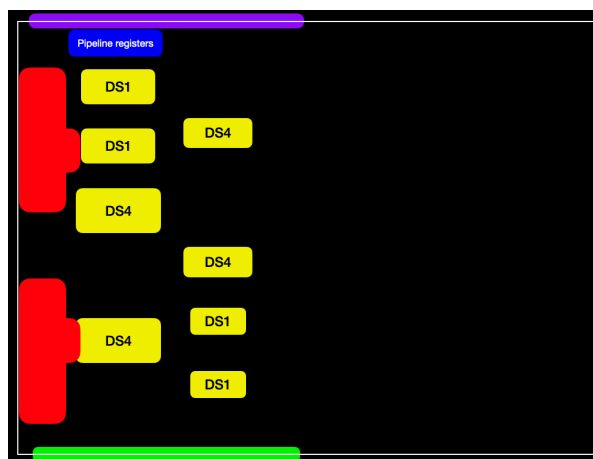


Figure 3.3.: 2 stages of DS1 and 2 stages of DS4

3. Feedthrough strategies

3.1. Normalized bus timing results

To assess the impact of the feedthrough strategies, the delay of the feedthrough was calculated. The following equation was used to calculate the delay -

$$delay = arrival_time_at_output_pin - startpoint_clock_latency \quad (3.1)$$

After that, the exact delay of the buffer stages was found, without the impact of clock latency and synthesis constraints. Then, they were normalized against the baseline. Each line on the feedthrough is treated as a data point, and box plots were made Figures. 3.4, 3.5, 3.6, 3.7 and 3.8 for certain timing scenarios comparing that scenario for all experiments.

The frequency has no impact in this assessment of the bus delay since this assessment is within one clock period, and the impact of everything other than the feedthrough path is being removed from the timing.

3.1.1. Scenario 1 - Low voltage and slew, room temperature, Cmax corner

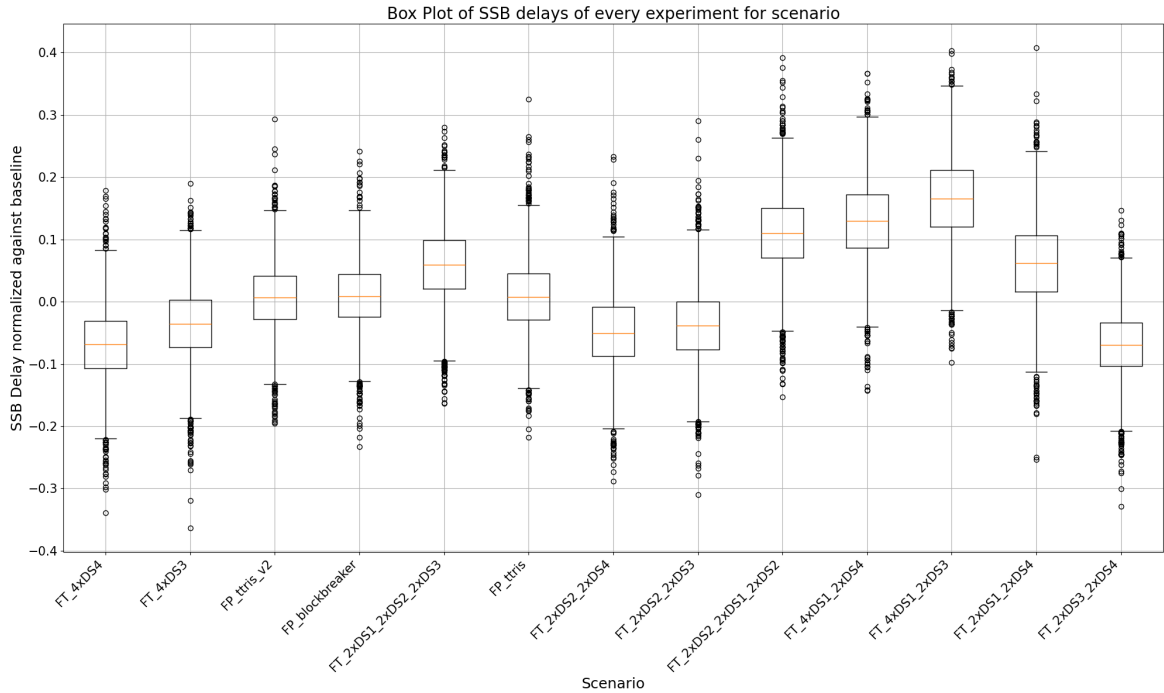


Figure 3.4.: Boxplot of low voltage and slew rate, room temperature, capacitive worst scenario

The first scenario is a low frequency and voltage, low slew rate at room temperature with the worst capacitive parasitics (cworst_ccworst). As the parameters of the scenario suggest, this is a slow corner. From the Figure 3.4, the following observations can be made-

3. Feedthrough strategies

- Experiments with a total of four stages with buffers of higher drive strength like DS3 and DS4 are faster than the baseline.
- The changes in the floorplan don't have a significant impact on the timing of the feedthrough.
- Experiments with slightly downsized buffers like the 2xDS2 2xDS1 2xDS2 and 4xDS1 2xDS3, expectedly have higher delays than baseline.

3.1.2. Scenario 2 - Low voltage and slew, cold temperature, crosstalk corner

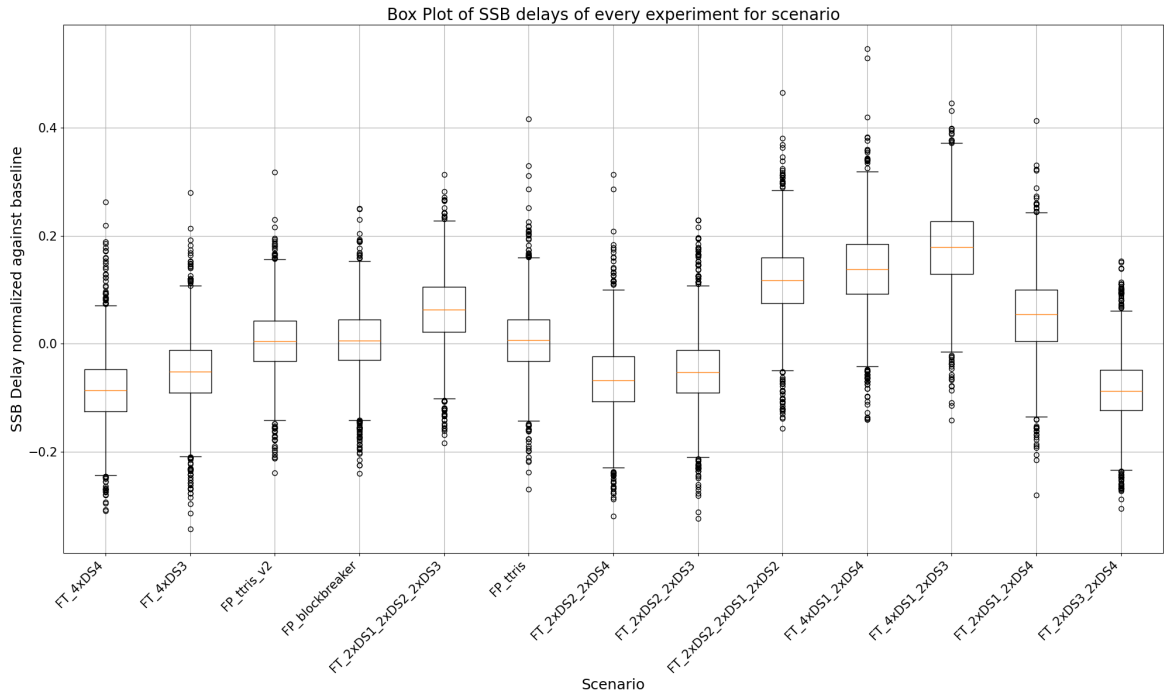


Figure 3.5.: Boxplot of low voltage and slew rate, cold temperature, crosstalk worst corner - scenario

This is a low frequency and voltage, low slew rate at cold temperature scenario with max coupling capacitance and min ground capacitance * resistance (rcbest_cbest). This is also a slow scenario, because of the low voltage and the frequency, and is dominated by crosstalk. Fig. 3.5 shows the box plot for this scenario, and the following observations can be made-

- Figure 3.5 shows experiments with four stages of higher drive strength buffers are faster.
- The height of the boxes is short, which means that the spread of the values of the delays is fairly uniform around the median.

3. Feedthrough strategies

3.1.3. Scenario 3 - Low voltage and slew, cold temperature, Cmax corner

The only difference between scenario 3 and the scenario 1 is the temperature, and between scenario 3 and scenario 2 is the parasitics corner. Hence, scenario 3 like scenario 1 and scenario 2 is a slow one, and following observations were made-

- The temperature does not affect the trends between the different experiments. But in absolute values it was observed that the colder temperature had lower delay.
- Between the Cmax corner and the crosstalk corner, there is not much difference between the normalized results but in absolute values it was observed that the Cmax corner had a lower delay. This means that the delay in the bus is crosstalk dominant. This is expected considering that these are long wires in a very close proximity to each other.

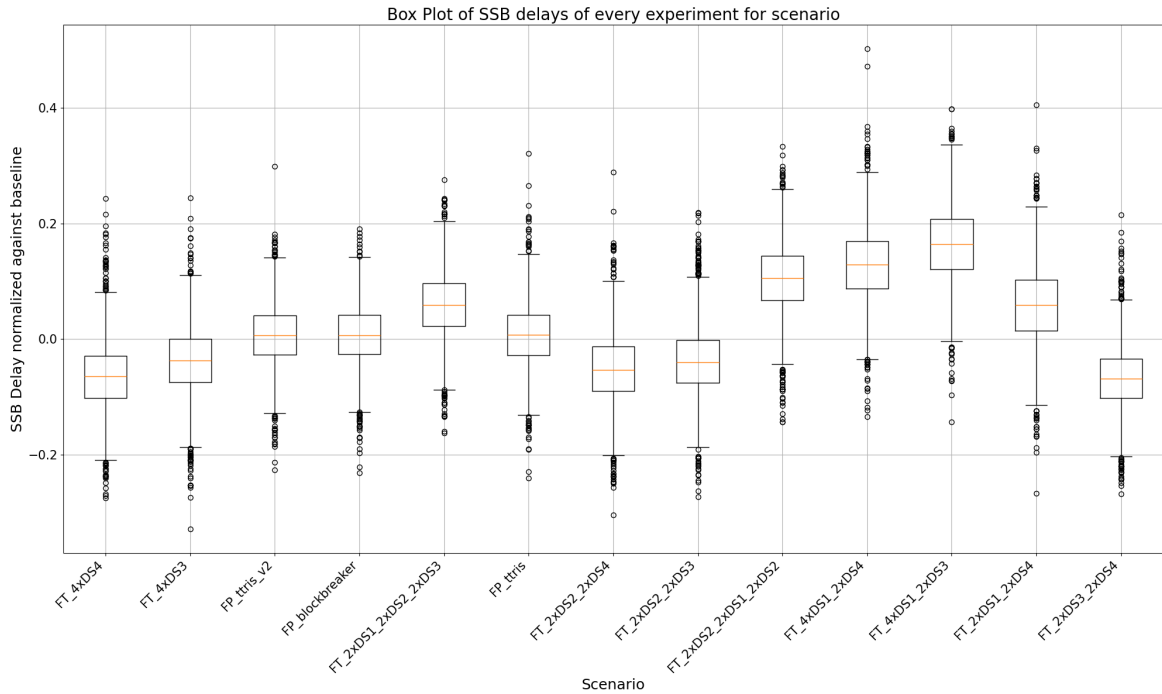


Figure 3.6.: Boxplot of low voltage and slew rate, cold temperature, capacitive worst scenario

3.1.4. Scenario 4 - High voltage and slew, cold temperature, crosstalk corner

This is a fast scenario given the high voltage and frequency. This scenario has some interesting observations and trends listed below-

- Here, four stages of high drive strength have higher delay than the baseline, which is a reversal from the trend in the slow corners.

3. Feedthrough strategies

- Again the experiments with lower drive strength buffers are also slower.
- The floorplan also affected the timing here. Upon close inspection, the height of the FP_blockbreaker experiment is lower. This is more observable in the absolute values for FP_ttris and FP_ttris_v2 experiments as well.
- The opposite of the above trend is also observed in experiments like 2xDS1 2xDS4, where the height of the box is taller indicating a higher spread in the values.

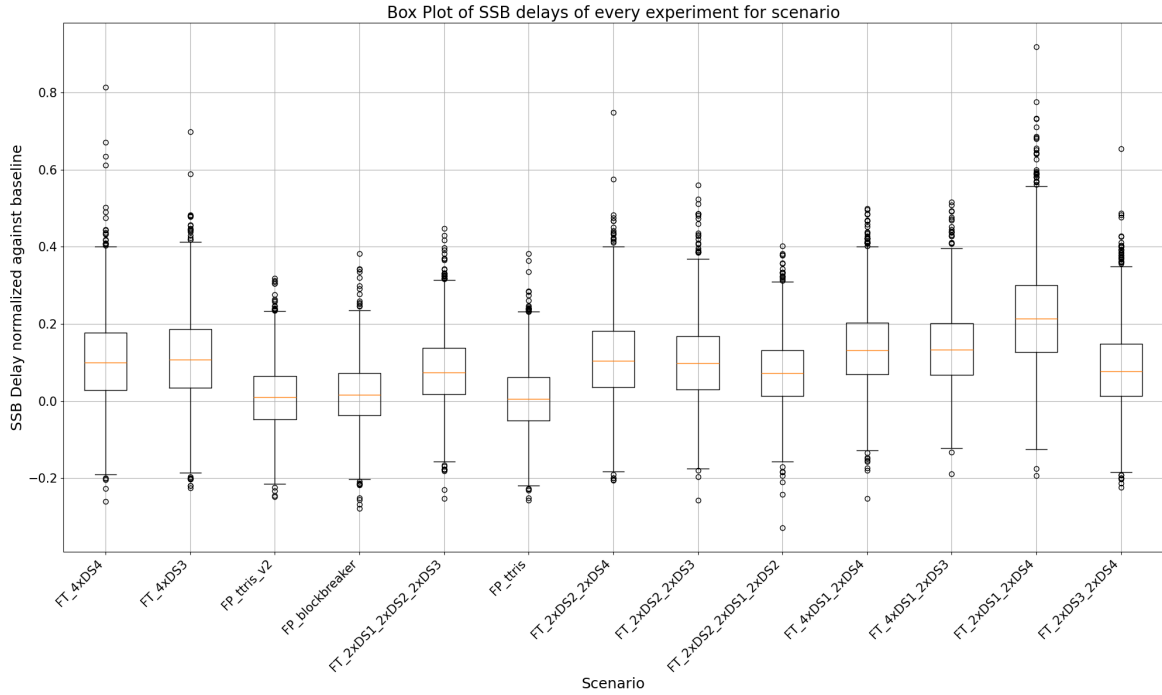


Figure 3.7.: Boxplot of High voltage and slew rate, cold temperature, crosstalk worst corner scenario

3.1.5. Scenario 5 - High voltage, low slew, cold temperature, Cmax corner

This is a fast-slow scenario since the frequency and voltage are high but the slew rate is low, and the only difference between scenario 5 and the scenario 3 is the voltage is high. It is observed that this scenario also follows the same trends as the fast scenario, indicating that the trends are dominated by the voltage rather than the slew rate.

Notice in the above plots, experiments FP_blockbreaker, FP_ttris, and FP_ttris_v2 are all nearly the same and close to the baseline, indicating that the strategy to pre-place the side inverters close to their fan in inverters to prevent it from impacting the feedthrough worked, and didn't impact the timing significantly.

3. Feedthrough strategies

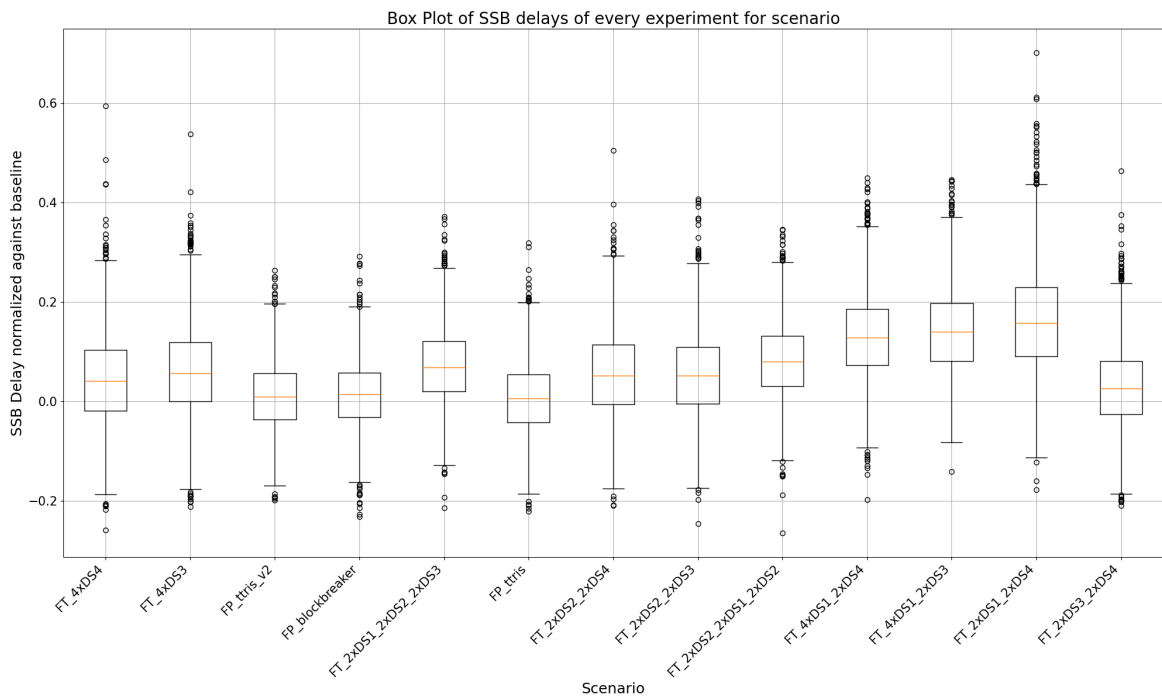


Figure 3.8.: Boxplot of High voltage, low slew rate, cold temperature, capacitive worst corner scenario

4. Results and Observations

4.1. Non-timing results

In Physical design, timing is an important parameter to evaluate a design. However, many other parameters are also important when evaluating the design, like power consumption, area, design rule checks (DRCs) etc. The following table shows the data normalized against the baseline for some of these important and relevant parameters.

NOTE: The results in the table are in percentage

Experiment	Route DRC	Route shorts	wirelength	Std cell area	Buff/Inv area	Leakage Power	Total Power
FP_blockbreaker	97.9	112.5	0.5	0.4	2.5	0.3	-0.8
FP_ttris	-21.4	-50.0	-0.1	-0.1	-1.7	-0.2	-1.4
FP_ttris_v2	-5.0	-4.2	-0.2	-0.3	-3.4	0.1	-1.6
4xDS4	-35.0	-29.2	-0.5	-0.5	-4.5	-0.6	-1.2
4xDS3	-15.0	-8.3	-0.7	-0.5	-5.0	-0.5	-1.8
2xDS2 2xDS3	4,014.3	2,700.0	1.9	-0.1	-3.3	1.5	-0.4
2xDS2 2xDS4	14,185.7	20,087.5	3.0	0.0	-3.8	3.6	0.0
2xDS3 2xDS4	3,960.7	2,145.8	3.3	0.2	-1.7	2.8	0.4
2xDS1 2xDS4	3,287.1	2,362.5	-0.5	-0.1	-0.3	-0.6	-1.6
2xDS2 2xDS1 2xDS2	3,179.3	2,279.2	0.8	-0.7	-7.9	0.0	-1.0
4xDS1 2xDS3	3,531.4	2,812.5	0.4	-0.6	-7.0	-0.1	-0.9
4xDS1 2xDS4	3,767.9	2,891.7	-0.6	-0.3	-0.3	-0.1	-1.7
2xDS1 2xDS2 2xDS3	4,725.0	3,379.2	1.4	0.1	-0.6	2.0	-0.4

Table 4.1.: Non-timing results

The results coloured in green are good since a reduction in everything is observed compared to the baseline. In contrast, the results in red are extremely bad since they bring minimal benefits in power consumption while increasing the cleanup effort significantly. From the table 4.1, it can be observed that all experiments with buffer stages of 2 different types have an exceptionally high number of DRCs and shorts. This could be because of how the placement of these buffers was implemented. In the feedthrough stage, a baseline of 4 DS4 or 6 DS2 stages is constructed, and marked don't touch. Then these baselines are resized and moved according

4. Results and Observations

to the desired configuration. Since these are marked as don't touch, in the script written to resize and move them, the argument to honor don't touch was disabled and then immediately after the resize and move, honor don't touch was turned back on, which prevented their move from being legalized, causing DRCs and shorts.

It can also be observed that the different floorplans lead to roughly 1 - 1.4% improvement in the power consumption for the partition, and this improvement is persistent even with the change in the buffering strategy. The anomalous cases are 2xDS2 2xDS4 and 2xDS3 2xDS4 which compared to 4xDS4 have 2 stages of buffers with lower drive strength, consumes same or more power as the baseline. The floorplan changes also lead to a reduction in area and in DRCs and shorts in FP_ttris and FP_ttris.v2 but an increase in for the same in FP_blockbreaker.

4.2. Timing results

This section discusses the impact of the floorplan and the buffering strategies on the overall timing of the partition.

Experiment	Setup R2R			Hold R2R		
	WNS	TNS	FEP	WNS	TNS	FEP
FP_baseline	-0.1714	-38.768	501	-0.1682	-78.8360	2254
FP_blockbreaker	-0.1087	-198.7296	1667	-0.0444	-75.3770	2466
FP_ttris	-0.1313	-27.4922	498	-0.0473	-65.7886	2667
FP_ttris.v2	-0.2243	-66.9870	784	-0.045	-56.0697	3646
4xDS4	-0.0841	-28.1678	467	-0.0401	-54.3719	2031
4xDS3	-0.0783	-46.7887	642	-0.0424	-66.7201	2067
2xDS2 2xDS3	-0.4784	-1144.3880	1460	-0.1981	-69.3855	2473
2xDS2 2xDS4	-0.7153	-1104.7171	2619	-0.1123	-104.4070	2599
2xDS3 2xDS4	-0.4216	-420.5976	1557	-0.1702	-64.8193	2508
2xDS1 2xDS4	-0.1079	-43.1404	473	-0.1053	-71.1467	2698
2xDS2 2xDS1 2xDS2	-0.1801	-44.4121	589	-0.0521	-67.0411	2824
4xDS1 2xDS3	-0.1066	-57.2617	1030	-0.1008	-71.0386	2233
4xDS1 2xDS4	-0.1923	-19.3940	468	-0.1904	-67.8024	2698
2xDS1 2xDS2 2xDS3	-0.1392	-61.4734	794	-0.1191	-73.6311	2463

Table 4.2.: Timing Results

From the above table, we can see that the experiments FP_ttris, 4xDS4 and 4xDS3 are improvements over the baseline in terms of Worst Negative Slack (WNS) and Total Negative Slack (TNS) for both setup and hold. Experiments like 2xDS2 2xDS3, 2xDS2 2xDS4, 2xDS3 2xDS4 are relatively worse for setup but not so bad for hold. The floorplan FP_blockbreaker has a lower WNS for setup but the TNS and Failing End Points (FEP) on the other hand are

4. Results and Observations

much worse compared to the baseline. Out of all the experiments, the best performer is 4xDS4 with the lowest numbers for all WNS, TNS, FEP for both setup and hold.

4.2.1. Why are the WNS and TNS numbers bad?

At first glance, the numbers for the WNS and TNS are not so great. This is because of a few reasons-

- The WNS numbers come from the constraints provided to the tool, and because these paths are critical feedthrough paths, they are over-constrained.
- These results are directly after PNR stage, and no timing fixing has performed on these results yet.

4.3. Future Scope

The whole Physical Design flow involves more analysis like EMIR, Noise, and Physical Verification. These were not performed in this project, but these are important analysis' that need to be performed before signoff and tapeout can happen. In this project feedthrough topologies with increasing buffer sizes were explored and not the topologies with decreasing buffer sizes. Assessing the impact of these different topologies might also be an interesting finding.

Bibliography

- [1] Shashank Prasad and Anuj Kumar. Simultaneous routing and feedthrough algorithm to decongest top channel. In *2009 22nd International Conference on VLSI Design*, pages 399–403, 2009.
- [2] Jin-Tai Yan. An optimal ilp formulation for minimizing the number of feedthrough cells in standard cell placement. In *Proceedings of the Sixth Great Lakes Symposium on VLSI*, pages 100–105, 1996.
- [3] Rajendra Bahadur Singh, Anurag Singh Baghel, and Ayush Agarwal. A review on vlsi floorplanning optimization using metaheuristic algorithms. In *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pages 4198–4202, 2016.
- [4] S.N. Adya, S. Chaturvedi, J.A. Roy, D.A. Papa, and I.L. Markov. Unification of partitioning, placement and floorplanning. In *IEEE/ACM International Conference on Computer Aided Design, 2004. ICCAD-2004.*, pages 550–557, 2004.
- [5] I. Hameem Shanavas and Ramaswamy Kannan Gnanamurthy. Wirelength minimization in partitioning and floorplanning using evolutionary algorithms. *VLSI Design*, 2011:896241, 2011.
- [6] C.F. Ball, P.V. Kraus, and D.A. Mlynski. Fuzzy partitioning applied to vlsi-floorplanning and placement. In *Proceedings of IEEE International Symposium on Circuits and Systems - ISCAS '94*, volume 1, pages 177–180 vol.1, 1994.
- [7] Ronak Lad (Einfochips Pvt. Ltd.) Pavan H Vora. A review paper on cmos, soi and finfet technology. Available at <https://www.design-reuse.com/articles/41330/cmos-soi-finfet-technology-review-paper.html>.