Report for the Power/Performance Estimations of GNN

1. Design Overview

For the RTL design, given the fact that deeper hidden layers have more bitwidth, which means more logic delay for multiplication and sum(), I added the DNN's flip flops stage with tendency to relieve deeper level's critical path. So as to realize performance/power comparison among different design option. I implement a RTL design of DNN with flexible bitwidth and flip flops stage insertion, which is shown in the MS1-MS5's submission. GNN designs in different frequency are implemented from behavioral to physical layout level. And finally they come to evaluation and comparison in this Milestone.

2. Design Comparison by PT

From the *reports_730 reports_883 and reports_970*, 2,4,full flip flop stages design's PT reports are included. The required metrics are listed below:

Design	Area(mm^2)	Max Frequency(MHz)	Min Latency(ns)	Power(mW)	Energy(pJ)	EDAP(pJ ns-mm^2)
full-stages	0.00244	970	7.22	6.89	49.75	0.876
4-stages	0.0022	883	4.53	5.79	26.23	0.261
2-stages	0.00197	726	2.75	3.86	10.61	0.057

From the Table we can see that the high-frequency design does poor at **latency**, instead it performs well in **throughput**, which this project spec doesn't cover, while lantency is quadratic to EDAP. Moreover, high-frequency design requires more areas for flip-flops and routing effort to meet timing closure. Meanwhile, high-frequency mode also denotes larger power. So considering the EDAP metric, the 2-stages design stands out. **It is important because further optimization is based on these findings from the initial designs and evaluation.**

3. Optimization and improved results

3.1. Outcome

According to files under *reports_opt724* folder. The improved results are listed below:

Design	Area(mm^2)	Max Frequency(MHz)	Min Latency(ns)	Power(mW)	Energy(pJ)	EDAP(pJ ns-mm^2)

Design	Area(mm^2)	Max Frequency(MHz)	Min Latency(ns)	Power(mW)	Energy(pJ)	EDAP(pJ ns-mm^2)
improved	0.00193	724	1.38	3.65	5.037	0.0134

The updated simplified design architecuture is shown in Figure 1:

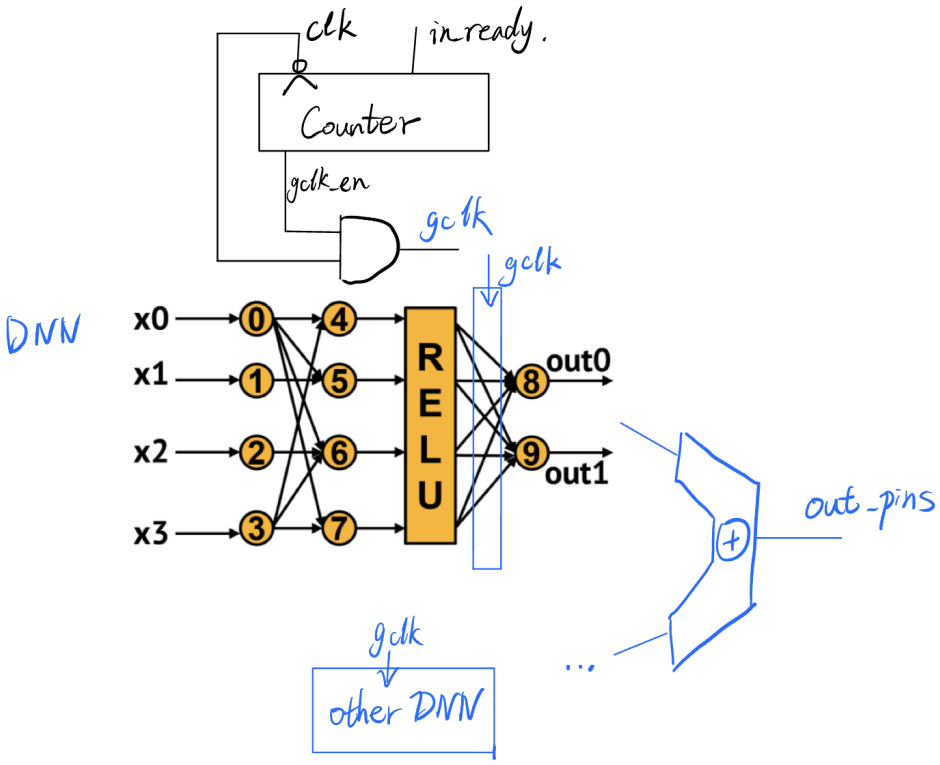

Figure 1. Simplified design architecture

3.2. Optimization

The optimization is based on the following two aspects:

3.2.1. Flip-flops stage

From Figure 1, we can see that only one stage of flip flops are inserted in the design which locates at deeper location so that it can effectively decrease critical path and latency cycle. Thus from the outcome table we can see that the latency is decreased from 2.74 to 1.38, meanwhile the frequency doesn't change too much. However, the configuration for APR is different so it may have some cost in area. However, less registers requirements also decrease the area. So the area is decreased from 0.00197 to 0.00193.

3.2.2. Gated Clock

From Figure 1, we can see that the gated clock is inserted in the design in order to decrease power. However, to implement the gated clk is not simply adding some logic. One tricky part is in the further Synthesis and APR.

1. In Synthesis and APR, we should add more timing constraints for analysis because it is now a multi-clock design. For example, I added the following constraints for DC shell:

```
# Find the destination pin of the clk_gated signal
set clk_gated_net [get_nets gclk]
set clk_gated_pin [get_pins -of_objects $clk_gated_net]
# Define gated clock
create_generated_clock -name clk_gated -source [get_ports clk] -divide_by 1
$clk_gated_pin

set_clock_groups -asynchronous -group [get_clocks clk] -group [get_clocks clk_gated]
```

To clarify that, I also attached the sdc file.

2. To avoid contention, from the figure 1 we can see that I generated the gclk_en by clk_negedge. Otherwise it will encounter hold time violation.

As a result, we can see a drop in power from 3.86 to 3.65, in the meantime the timing analysis is still passed.

3.3. Validation

Since I have changed the design, I require to go through MS2-MS5 again to validate the design and generate outputs. In short, here I just show the post-simulation results similar in MS3. Other internal reports have been uploaded to github repo. If they are required, please check the *optimized_* folders.

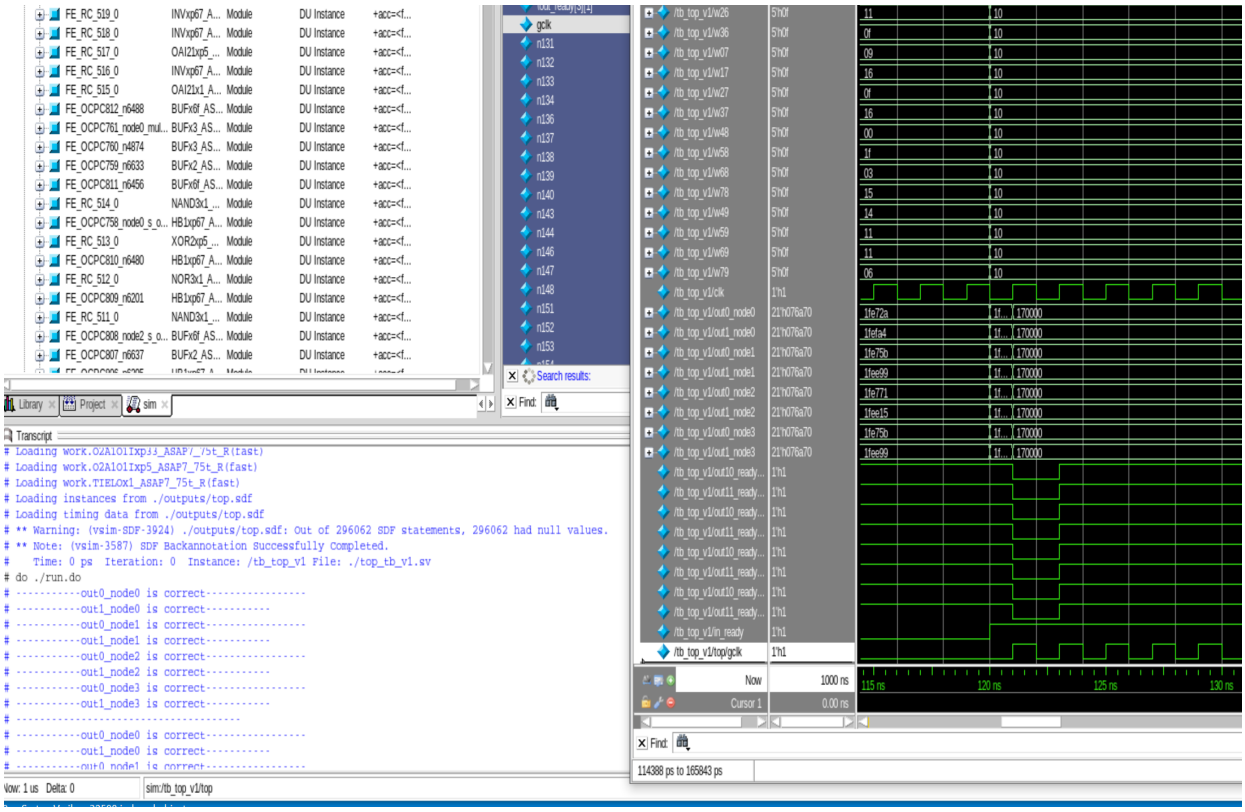

Figure 2. Post-simulation waveform

As shown in the waveform, the latency is 1 clk, the clock is gated and all outputs are correct.

4. Further Discussion

1. About the verification part, I have generated the .sdf by PrimeTime for modelsim to do backannotation. While it seems that modelsim cannot identify these parastic delay. Is it out of version compatibility? Moreover, I cannot open the VCS on CAE machine which obstruct me from further verification, as we can see in Figure 3.
2. Further optimization option. In this design, 4 DNNs are implemented for GNN. However, DNNs can be shared to save area. It is a common shrinking method implemented by some more timing control and results latched logic, but it will worsen the lantency.
3. Limitation: this design has no rst signal from the spec. So it cannot maintain some more complicated logic safely.