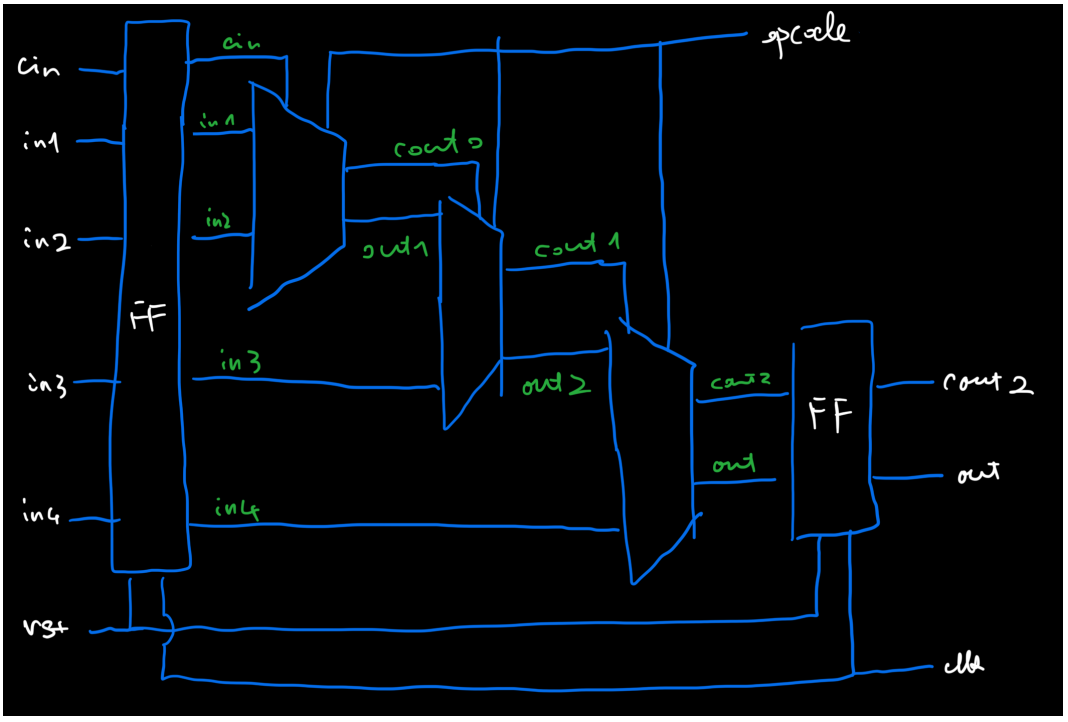
Report Lab 3

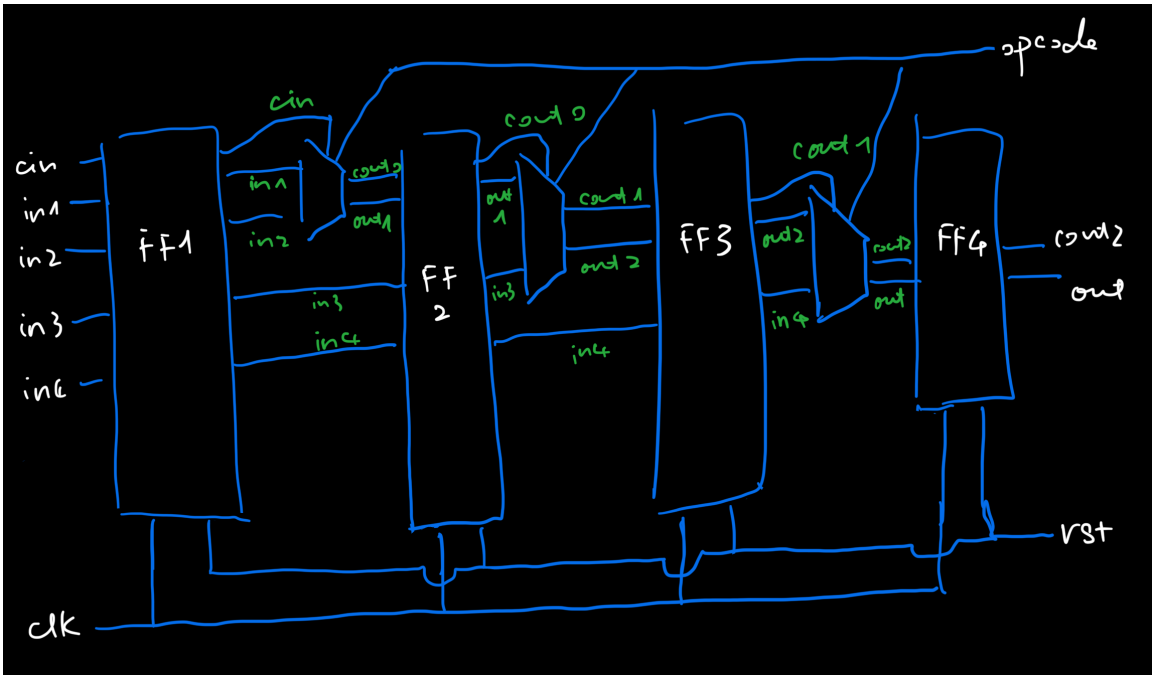
In this report the series ALU and pipeline ALU are compared

1. Block diagram of two different implementations

1. Series ALU (non-pipeline):



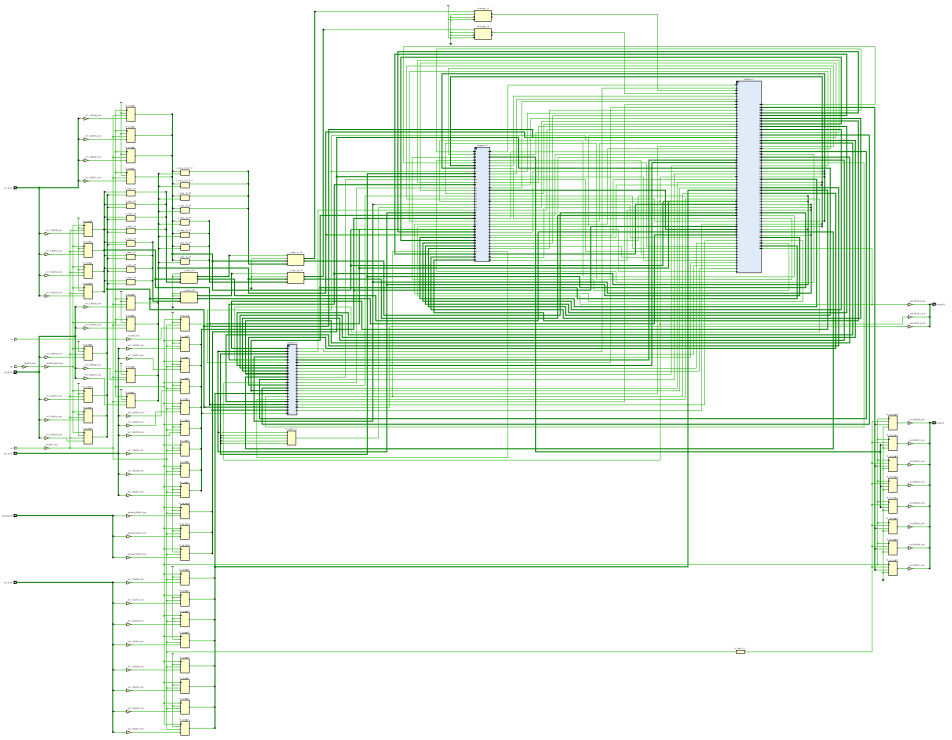
2. Pipeline ALU:



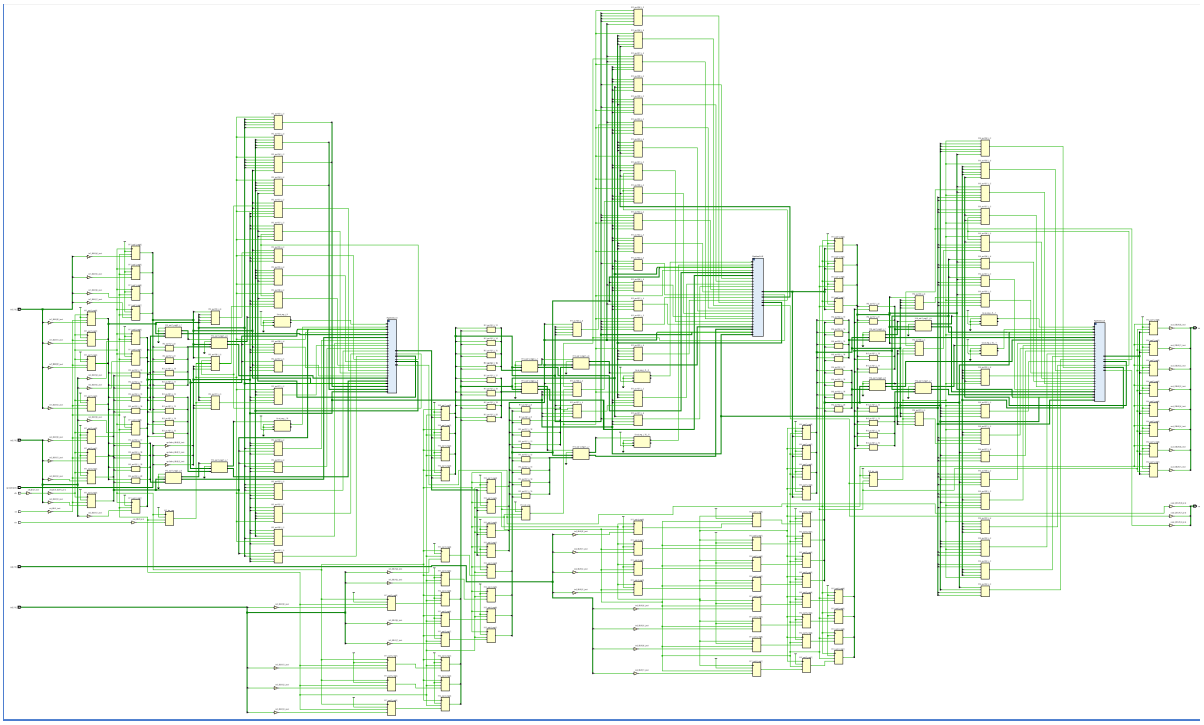
We can find that the differences are that for pipeline ALU, couple of FFs care inserted in between the ALUs. The architecture is more complicated. And for non-pipeline ALU, in order to introduce timing into it, we need to add 2 extra FF sets at 2 sides of the circuit.

2. Schematic after synthesis

1. Series ALU (non-pipeline):



2. Pipeline ALU:



Also, from the schematic after the synthesis, we can see that lots of FFs are inserted in between the ALUs for pipeline, and set of FFs are added at the input and output of the circuit.

3. Area comparison

1. Series ALU (non-pipeline):

Site Type	Used	Fixed	Available	Util%
Slice LUTs	263	0	20800	1.26
LUT as Logic	263	0	20800	1.26
LUT as Memory	0	0	9600	0.00
Slice Registers	47	0	41600	0.11
Register as Flip Flop	44	0	41600	0.11
Register as Latch	3	0	41600	<0.01
F7 Muxes	0	0	16300	0.00
F8 Muxes	0	0	8150	0.00

2. Pipeline ALU:

Site Type	Used	Fixed	Available	Util%
Slice LUTs	261	0	20800	1.25
LUT as Logic	261	0	20800	1.25
LUT as Memory	0	0	9600	0.00
Slice Registers	86	0	41600	0.21
Register as Flip Flop	83	0	41600	0.20
Register as Latch	3	0	41600	<0.01
F7 Muxes	0	0	16300	0.00
F8 Muxes	0	0	8150	0.00

For the area report after implementation, we can find that the number of LUTs are almost the same, this is reasonable because both ALU consists of 3 same simple ALUs, which can sure to result in a same LUTs. The amount of FFs is quite different, since for series ALU, FFs are only added at the input and output of the circuit while for the pipeline ALU, FFs are also added in between the simple ALUs, which can result in more FFs.

Therefore, according to the area report, we can say that the are of the non-pipeline ALU is smaller than pipeline ALU.

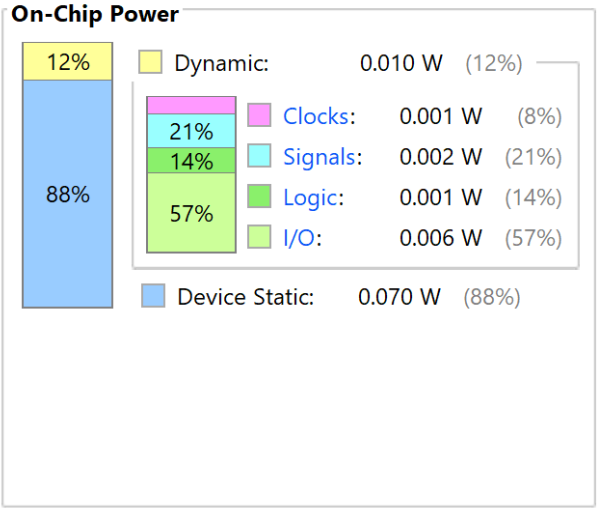
4. Power comparison:

1. Series ALU (non-pipeline):

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	**0.08 W**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**25.4°C**
Thermal Margin:	59.6°C (11.9 W)
Effective ϑJA:	5.0°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

Launch Power Constraint Advisor to find and fix invalid switching activity

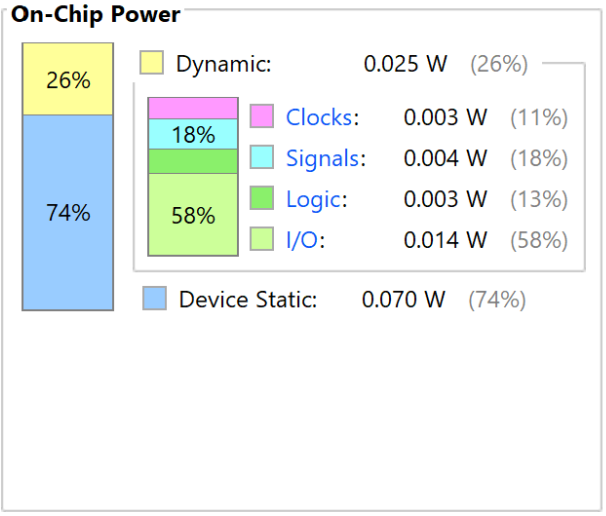


2. Pipeline ALU:

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power:	**0.095 W**
Design Power Budget:	**Not Specified**
Power Budget Margin:	**N/A**
Junction Temperature:	**25.5°C**
Thermal Margin:	59.5°C (11.8 W)
Effective ϑJA:	5.0°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

Launch Power Constraint Advisor to find and fix invalid switching activity



According to the power report, we can find that the total on-chip power of the pipeline ALU is a little bit higher than the power of non-pipeline ALU, this is also straightforward since for pipeline ALU, more components is used as shown in the are report, which is sure to have a relatively higher power consumption compared with the non-pipeline ALU.

5. Performance comparison:

1. Series ALU (non-pipeline):

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	0.696 ns	Worst Hold Slack (WHS):	0.544 ns	Worst Pulse Width Slack (WPWS):	5.500 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	8	Total Number of Endpoints:	8	Total Number of Endpoints:	45

All user specified timing constraints are met.

Period is 12ns

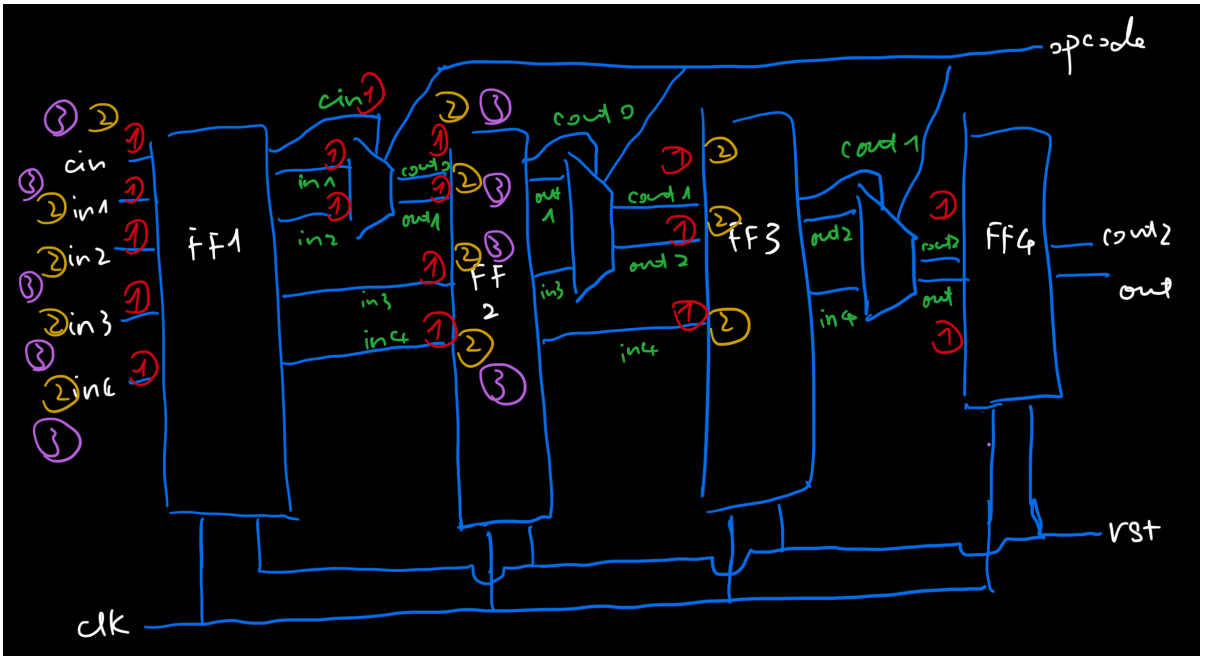
2. Pipeline ALU:

Setup		Hold		Pulse Width	
Worst Negative Slack (WNS):	0.492 ns	Worst Hold Slack (WHS):	0.200 ns	Worst Pulse Width Slack (WPWS):	2.000 ns
Total Negative Slack (TNS):	0.000 ns	Total Hold Slack (THS):	0.000 ns	Total Pulse Width Negative Slack (TPWS):	0.000 ns
Number of Failing Endpoints:	0	Number of Failing Endpoints:	0	Number of Failing Endpoints:	0
Total Number of Endpoints:	48	Total Number of Endpoints:	48	Total Number of Endpoints:	84

All user specified timing constraints are met.

Period is 5ns

According to the WNS shown in the performance comparison, we can see the the max allowed frequency is quite different. The max frequency for a non-pipeline ALU can only be around 80MHz (12ns), while for pipeline ALU, it could reach 200MHz, which is much higher than the non-pipeline one. The result can be explained as the data flow in pipeline ALU shown as below:



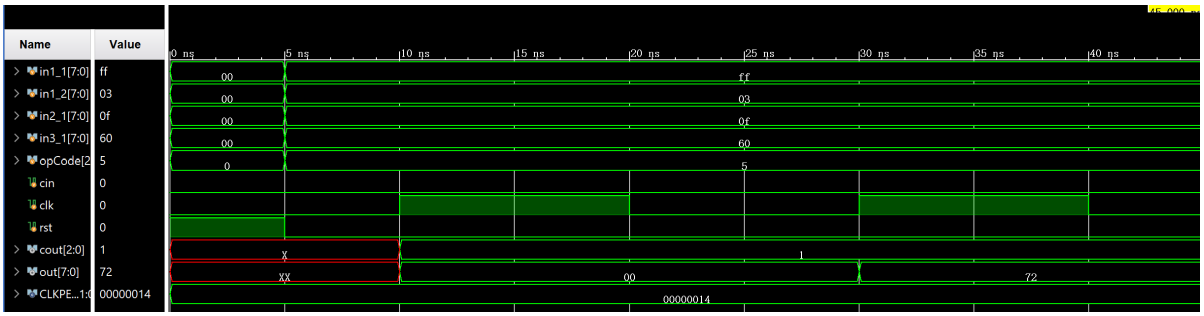
The data flow for pipeline ALU would be

1. the first data (denotes as 1 in red) arrives at FF1.
2. at posedge clk, data1 reaches FF2. At this time the data2 (denotes in yellow) can come to FF1 and wait for the posedge clk.
3. at posedge clk, data1 reaches FF3, data2 reaches FF2 and meanwhile, data3 (denotes in purple) can come to FF1 and wait.
4. ....

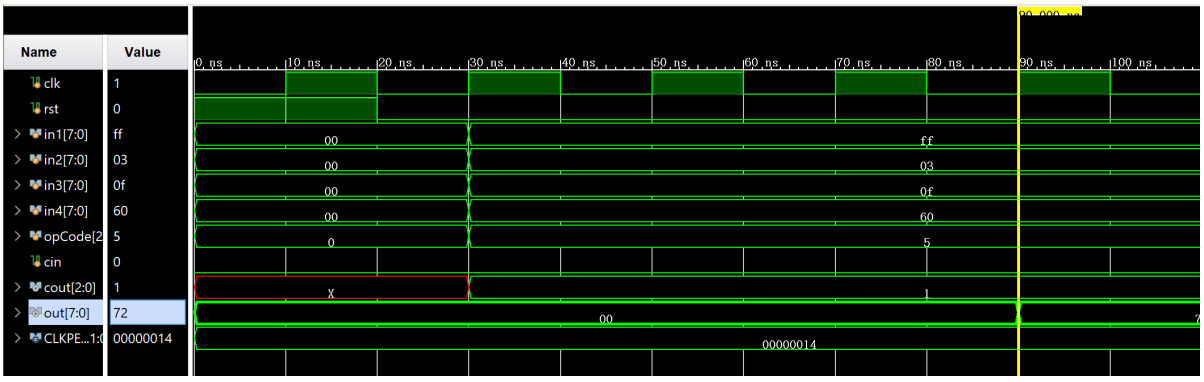
This means that, when using pipeline ALU, data1, data2 and data3 are isolated from each other. Like they are processed individually. Because at each poesdge clk, the previous data has gone to the next FF and been 'isolated' by this FF, in this case the next data can come and process.

6. Waveform

1. Series ALU (non-pipeline):



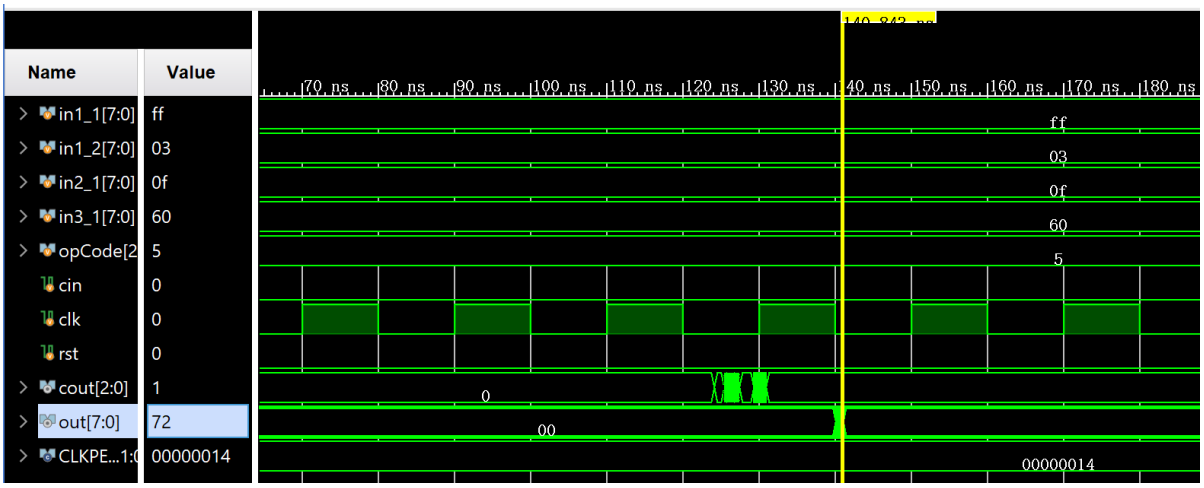
2. Pipeline ALU:



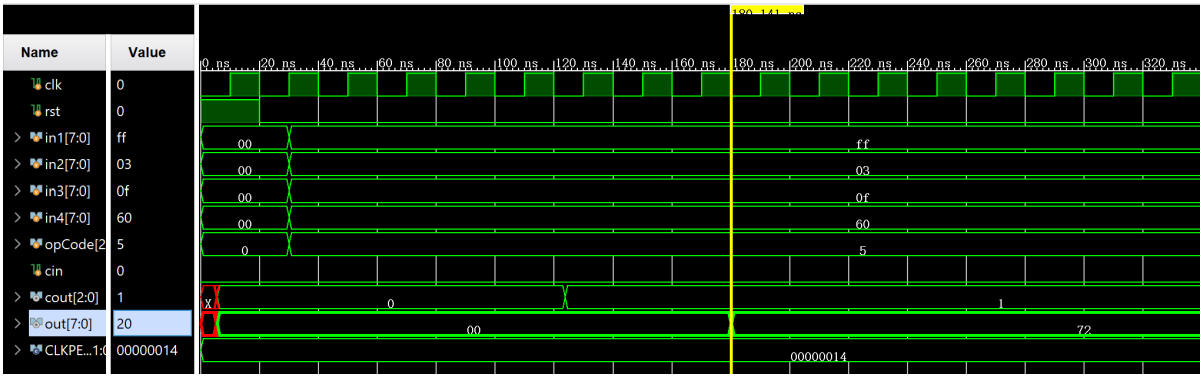
From the behavior simulation above, we can further validate our pipeline design. For the non-pipeline, since FFs are only added at the input and output, the result can be then obtained after 2 posedge clk. For the pipeline ALU, apart from the FFs at the input and output, FFs are also inserted in between the ALUs, which means that 3 clk cycles are needed to obtain the result.

Then we can run the post-implement timing simulation:

1. Series ALU (non-pipeline):



2. Pipeline ALU:



We can find that the delay of the pipeline ALU is severe than the delay of the non-pipeline ALU, this is because in pipeline ALU, there are more FFs, which may cause higher delay.

7. Conclusion for PPA

	Power	Performance	Area
Pipeline ALU	0.095W	≈ 200MHz	#FF = 83, #LUT = 261
Non-pipeline ALU	0.08W	≈ 80MHz	#FF = 44, #LUT = 263

Therefore, pipeline ALU has higher power consumption, larger area but way better performance. Non-pipeline ALU has lower power consumption, smaller are but also much worse performance.

If the number of the stages is very large (or even inf) -> *pipeline is recommended*. Since though using the pipeline will result in higher power consumption and area, we can still make the max operating frequency at a relatively high level. If we use non-pipeline, than it would take really really long time to get a result, which is a bad idea though the power consumption and area are better.