

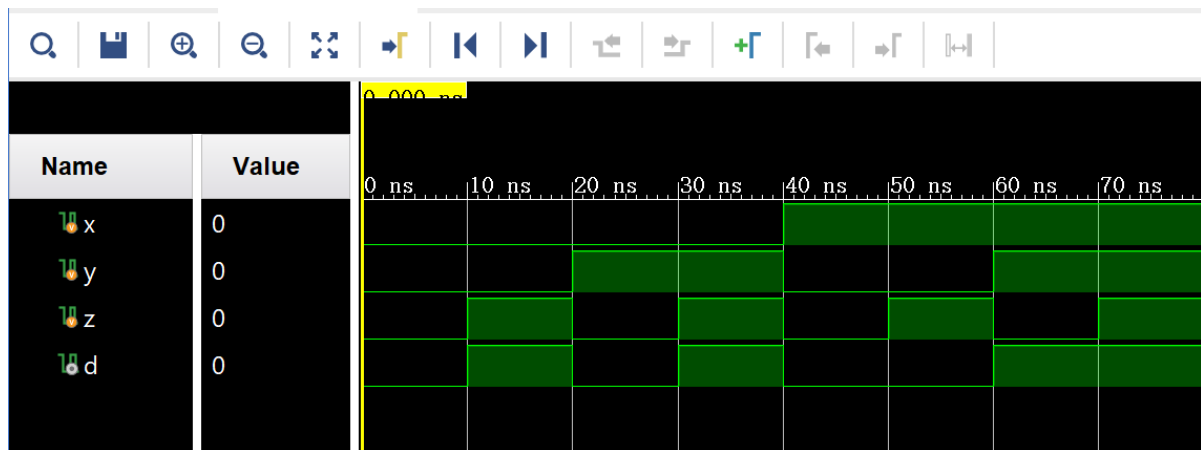
# Report Lab 1

In this report I focus on design flows: how to go from RTL to P&R and compare the differences between waveform, power report, schematic and so on.

## 2.1 Simple Combinatorial Circuit CC\_11

### Step 1: Behavior Simulation

After the RTL design of the circuit is finished, first run the simulation in order to have a initial check of the logic, shown as below:



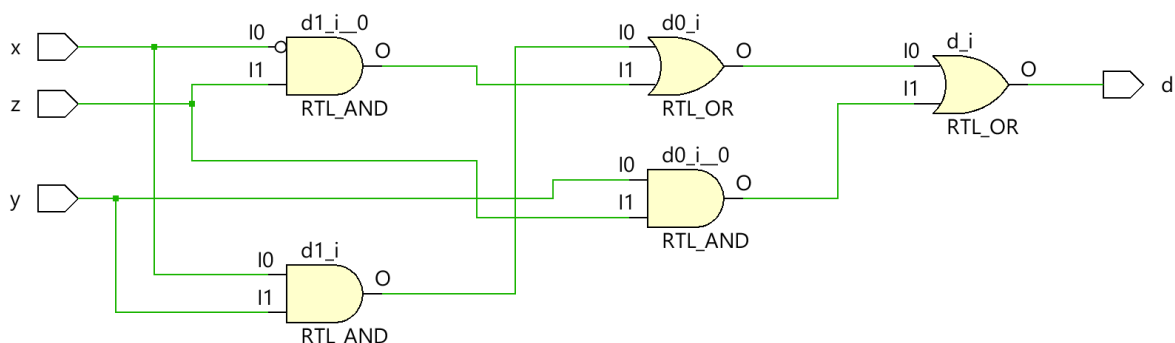
The waveform in simulation step will tell you if the logic of your design is correct. If this is correct, then go synthesis:

### Step 2: Synthesis

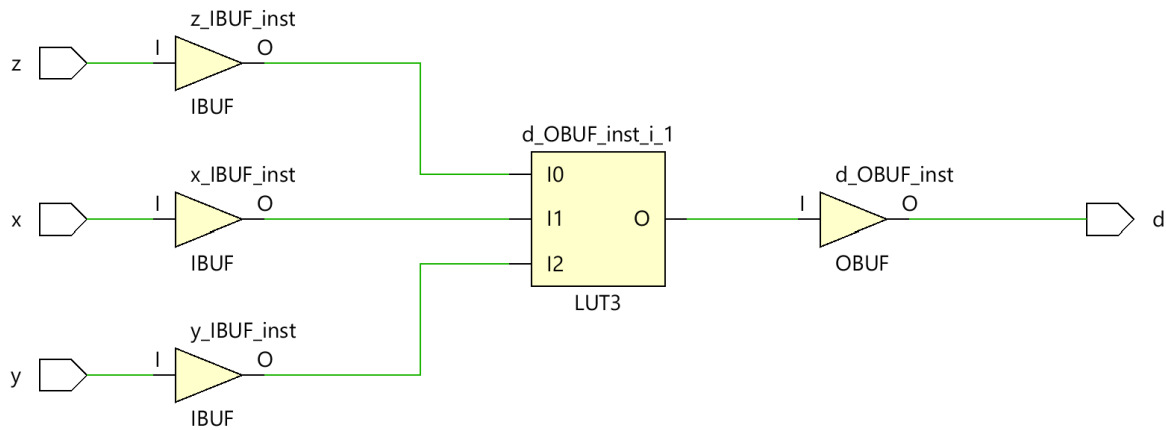
----- It provides technology dependent and somehow optimal (logic) view of your design. However the physical implementation properties are only estimated since the design is not routed yet. (From H409 CH5)

This is a more realistic step than simulation: in step 1 we wrote the abstract RTL design and now we will convert this "description" of the circuit into a more realistic netlist, which contains the real gate logic, FFs...

And actually the so-called **synthesis** is done in 2 steps: first elaboration and then synthesis. Therefore run the elaboration first:



and the Verilog description of the circuit is converted into more realistic logic gates. Then run the synthesis:



It is obvious that the schematic is quite different from the elaborating. This is because the elaborating is solely the analysis of the RTL code, but for the synthesis, it is the real existing unit and component from Xilinx (or other company), for example LUTs, IBUF... Also pay attention to the buffer that added to the input and output. Now it is more like a real "RTL" structure (*a set of input (source) and output (destination) FFs with gates or LUTs in the cloud in between* as explained in slides).

The power report for the synthesis step is :

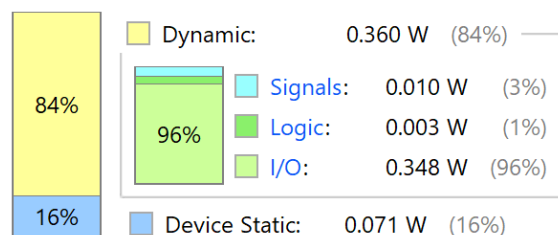
#### Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

**Total On-Chip Power:** 0.431 W  
**Design Power Budget:** Not Specified  
**Power Budget Margin:** N/A  
**Junction Temperature:** 27.2°C  
 Thermal Margin: 57.8°C (11.5 W)  
 Effective  $\theta_{JA}$ : 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

#### On-Chip Power



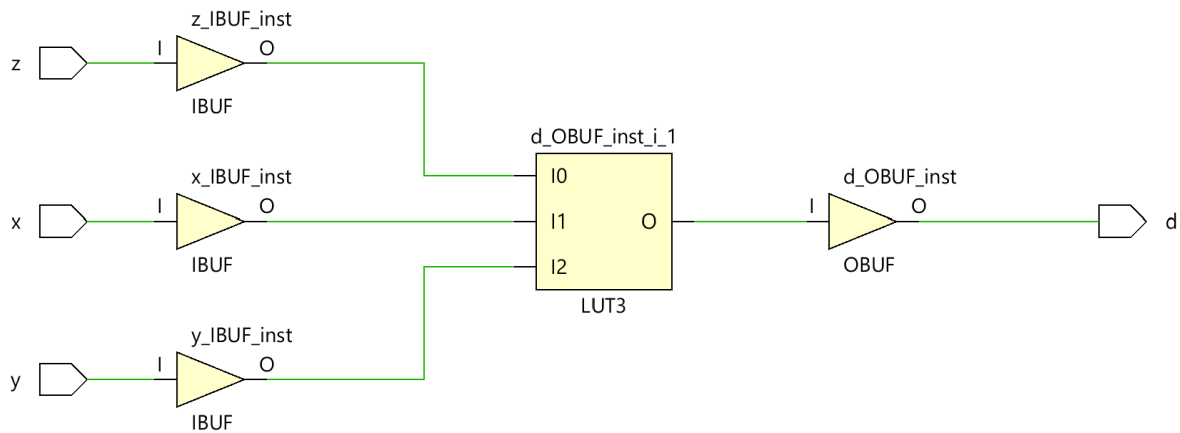
The power has dynamic (due to switching) and static (due to leakage). And it is worth noticing that the largest portion of power is consumed in I/O. This is because this is a small circuit (small area), the area of the I/O in this case is way larger than the area of the logic, leading to a much higher power consumption. After **synthesis**, the place and route (P&R) should be implemented, i.e. **implementation**. Since this is a purely combinatorial circuit, no timing report here.

### Step 3: Implementation

In this step, the synthesized netlist in step 2 will be further placed and routed for the physical implementation. Logically it is in 2 steps:

1. *Placement*: Placement engine positions cells (LUTs) from the netlist onto specific sites in the target FPGA device (x, y locations).
2. *Routing*: All pins (terminals) in the design are connected using physical nets . Pins = all IO & intermediate logic, LUTs, memory, DSPs...

The schematic from implementation step:



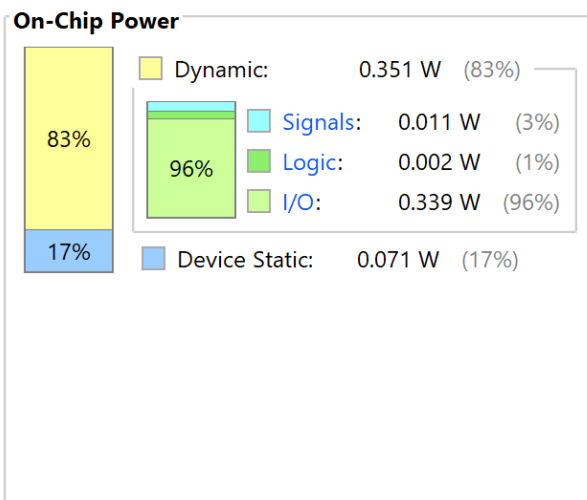
For this circuit, the schematic of synthesis and implementation is quite similar. Again, since this is a very, very simple circuit, it is normal that the schematic won't change a lot between this 2 steps. However, when looking at the power report, the result may change a little bit:

### Summary

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

<b>Total On-Chip Power:</b>	<b>0.422 W</b>
<b>Design Power Budget:</b>	<b>Not Specified</b>
<b>Power Budget Margin:</b>	<b>N/A</b>
<b>Junction Temperature:</b>	<b>27.1°C</b>
Thermal Margin:	57.9°C (11.5 W)
Effective $\theta_{JA}$ :	5.0°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low

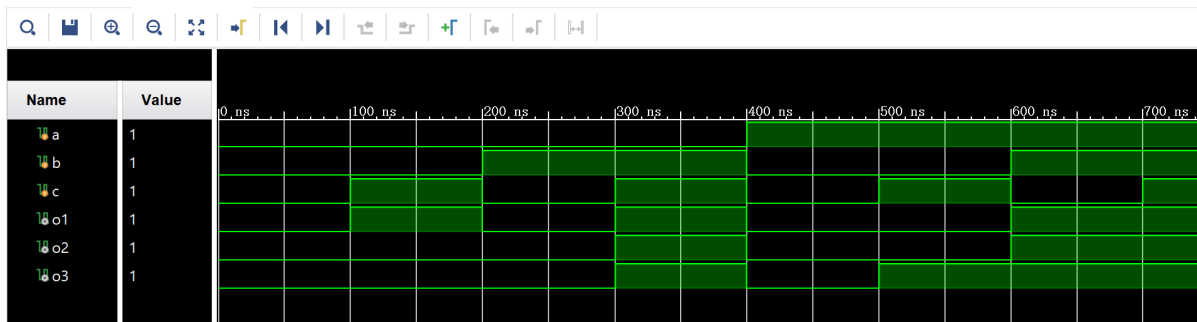
[Launch Power Constraint Advisor](#) to find and fix invalid switching activity



In dynamic power, all of signals, logic and I/O power changes a little bit. This is because this is in implementation step, the location of the circuit (each components) are placed and connected via "wires". This is the real physical implementation, so the power consumption in this step is the closest to the real case.

## 2.2 Top Level of CC\_11

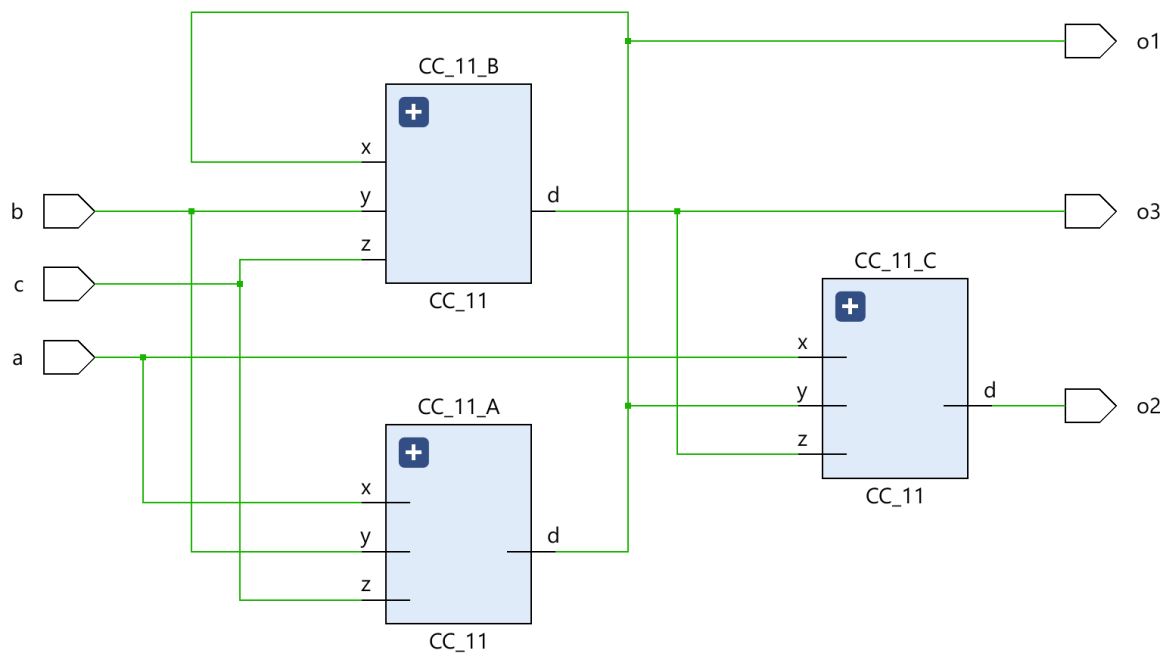
### Step 1: Behavior simulation:



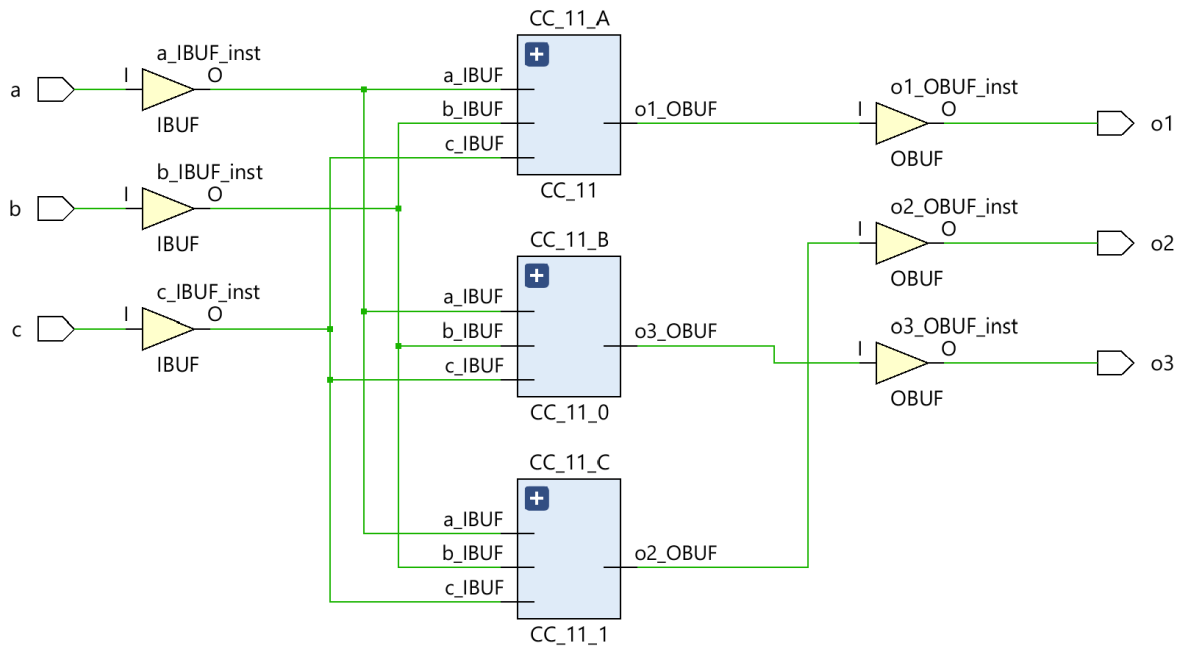
## Step 2: Synthesis

For this circuit it is quite interesting. Since during the lab session I find that the schematic generated from synthesis is quite different from what I expected (since I assume some inputs of the second and third CC\_11 will come from the output of the first CC\_11, however in reality it is not the case. See the comparison below)

After doing the elaborating step, as explained in 2.1, it will generate the schematic purely according to the RTL design, so it generates exactly the same schematic as I expected (pay attention to the input and output of the CC\_11):

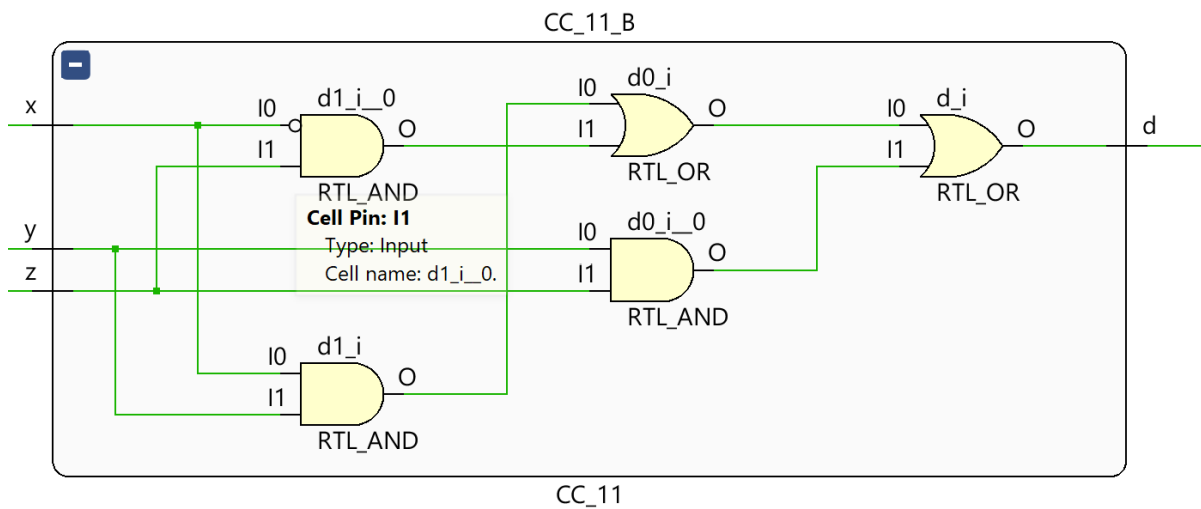


After elaborating, the synthesis begins. And see the schematic of synthesis step, it is now quite different from the elaborating one.

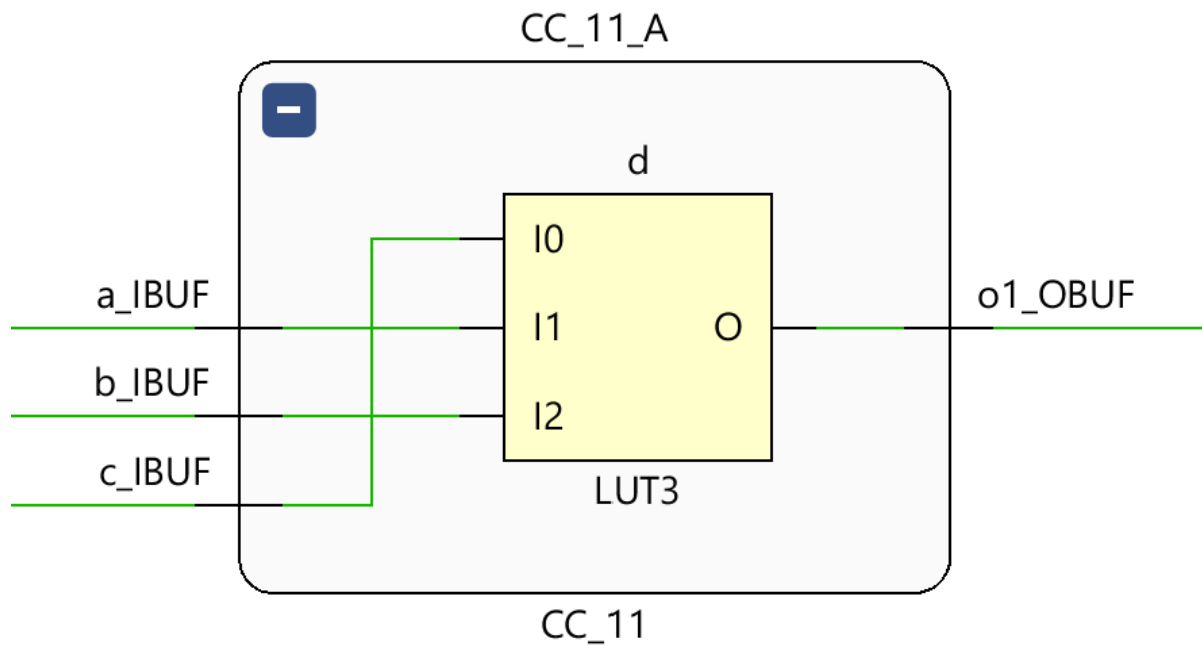


Apart from the input & output buffers, the way of connection also changes. This might be due to the fact that in synthesis step, some optimizations are done as well. It is also interesting to look into the CC\_11 blocks in both elaborating step and synthesis step and compare them.

- block details for elaborating step. We can find that the RTL description of the circuit is totally converted into set of gates:



- block details for synthesis step. We can find that the gates in elaborating step is converted to the existing LUT modules in Vivado, which is somehow optimized.



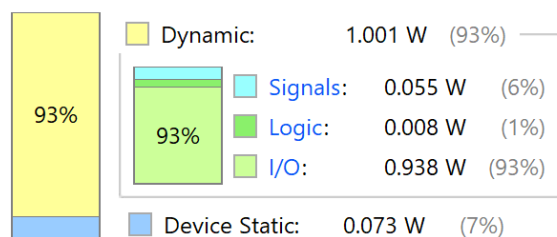
The power report is as follow, and the portion of each part is still quite similar with the CC\_11, which can be explained by using the same explanation as in 2.1.

### Summary

Power estimation from Synthesized netlist. Activity derived from constraints files, simulation files or vectorless analysis. Note: these early estimates can change after implementation.

<b>Total On-Chip Power:</b>	<b>1.074 W</b>
<b>Design Power Budget:</b>	<b>Not Specified</b>
<b>Power Budget Margin:</b>	<b>N/A</b>
<b>Junction Temperature:</b>	<b>30.4°C</b>
Thermal Margin:	54.6°C (10.9 W)
Effective $\theta_{JA}$ :	5.0°C/W
Power supplied to off-chip devices:	0 W
Confidence level:	Low
<a href="#">Launch Power Constraint Advisor</a> to find and fix invalid switching activity	

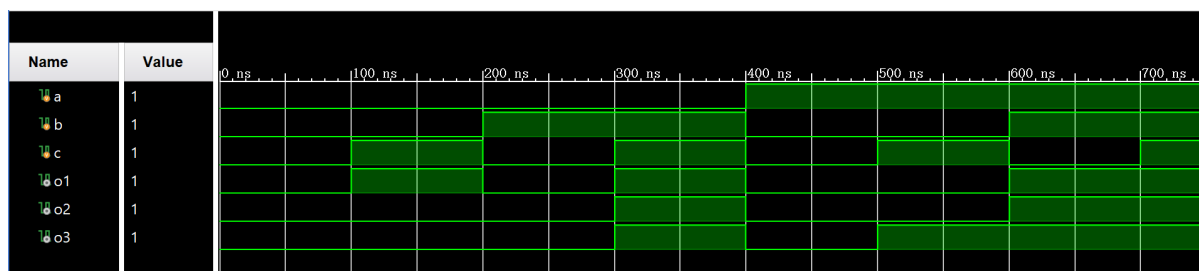
### On-Chip Power



The area can be checked in utilization report. 3 LUTs are used as expected.

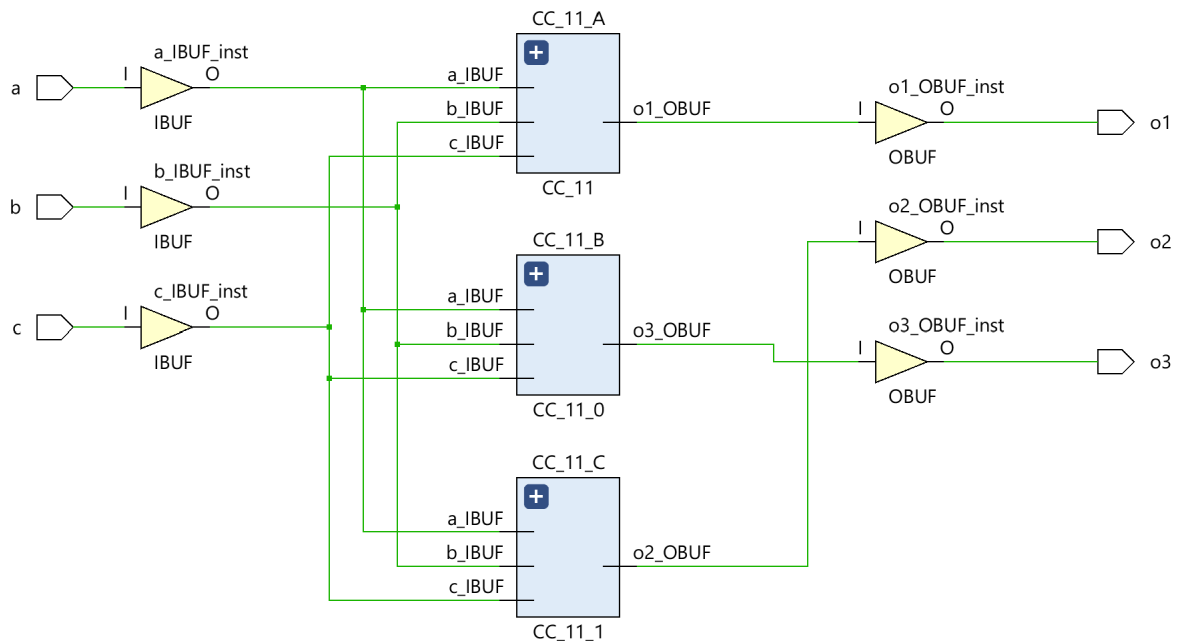
Site Type	Used	Fixed	Available	Util%
Slice LUTs*	3	0	20800	0.01
LUT as Logic	3	0	20800	0.01
LUT as Memory	0	0	9600	0.00
Slice Registers	0	0	41600	0.00
Register as Flip Flop	0	0	41600	0.00
Register as Latch	0	0	41600	0.00
F7 Muxes	0	0	16300	0.00
F8 Muxes	0	0	8150	0.00

After synthesis, we can run **post-synthesis simulation** to check if the timing and function of the circuit remains the same. There will be 2 different simulations which are timing simulation and functional simulation. The timing simulation considers the delay of the signal while functional simulation focus on the function after synthesis. The result is as below and it is the same as RTL simulation. It is good since this means that the synthesis of the circuit does not influence the behavior of the original design. The timing simulation will be shown at the end of the report in order to compare it with post-implementation and behavior simulations.



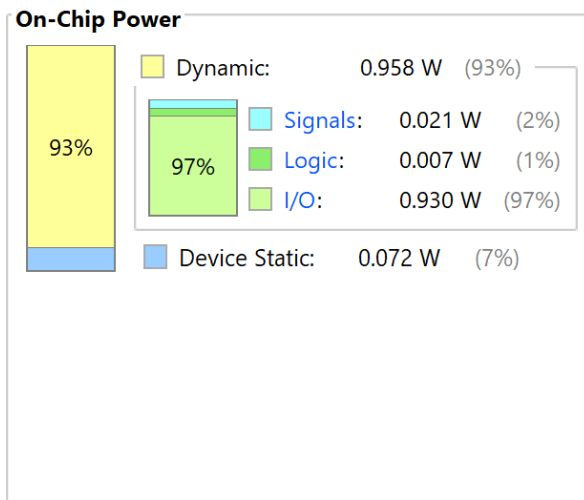
### Step 3: Implementation

After the implementation, the circuit is "mapped" to the real physical implementation. The schematic and power report is shown as below:

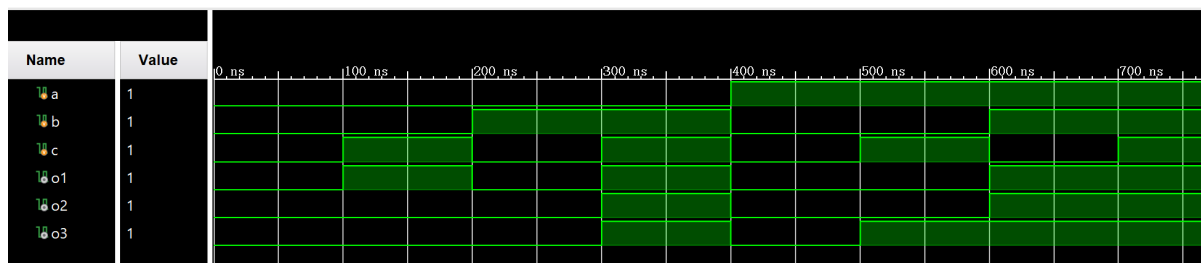


Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

**Total On-Chip Power:** 1.03 W  
**Design Power Budget:** Not Specified  
**Power Budget Margin:** N/A  
**Junction Temperature:** 30.2°C  
 Thermal Margin: 54.8°C (10.9 W)  
 Effective  $\theta_{JA}$ : 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



Again, small differences in power between synthesis step and implementation step due to P&R. Then do the **post-implementation** functional simulation:



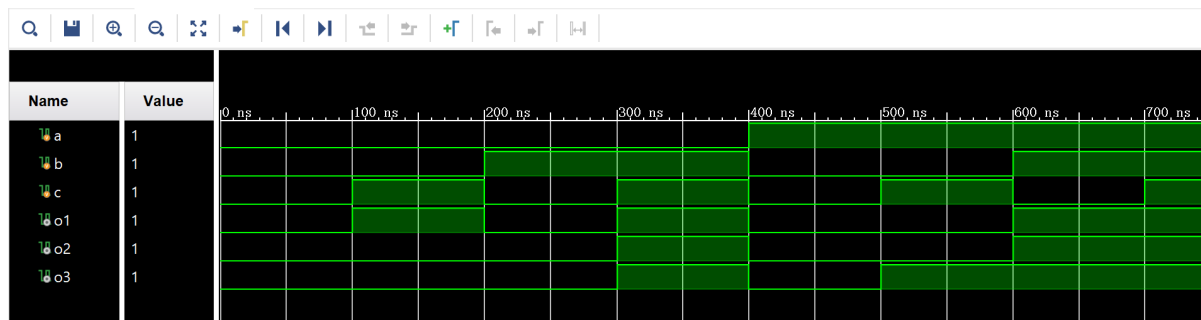
The function of the circuit remains the same with the presence of the input and output buffers and also P&R!

The area report is shown as below, which is the same as synthesis one:

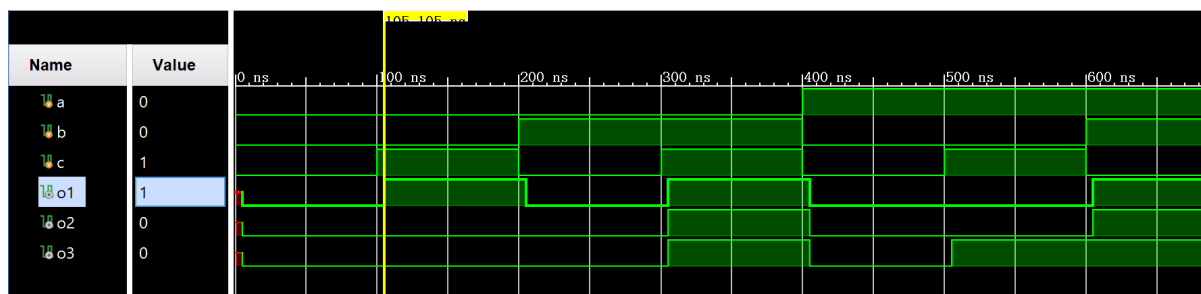


Site Type	Used	Fixed	Available	Util%
Slice LUTs	3	0	20800	0.01
LUT as Logic	3	0	20800	0.01
LUT as Memory	0	0	9600	0.00
Slice Registers	0	0	41600	0.00
Register as Flip Flop	0	0	41600	0.00
Register as Latch	0	0	41600	0.00
F7 Muxes	0	0	16300	0.00
F8 Muxes	0	0	8150	0.00

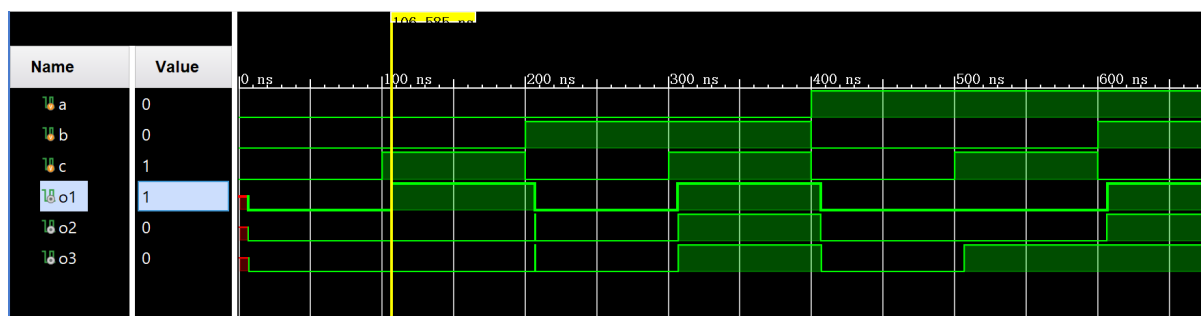
Comparison between behavior simulation, post-synthesis timing simulation and post-implementation timing simulation:



behavior simulation



post-synthesis timing simulation



post-implementation timing simulation

We can find that it has delays in the waveform, and the delay of the post-implementation timing simulation is always longer than the post-synthesis timing simulation. This could be easy to understand because after P&R, the length of the wire (or the path that signal should travel through) will be longer, which leads to a longer delay.

Might be a GOOD circuit...