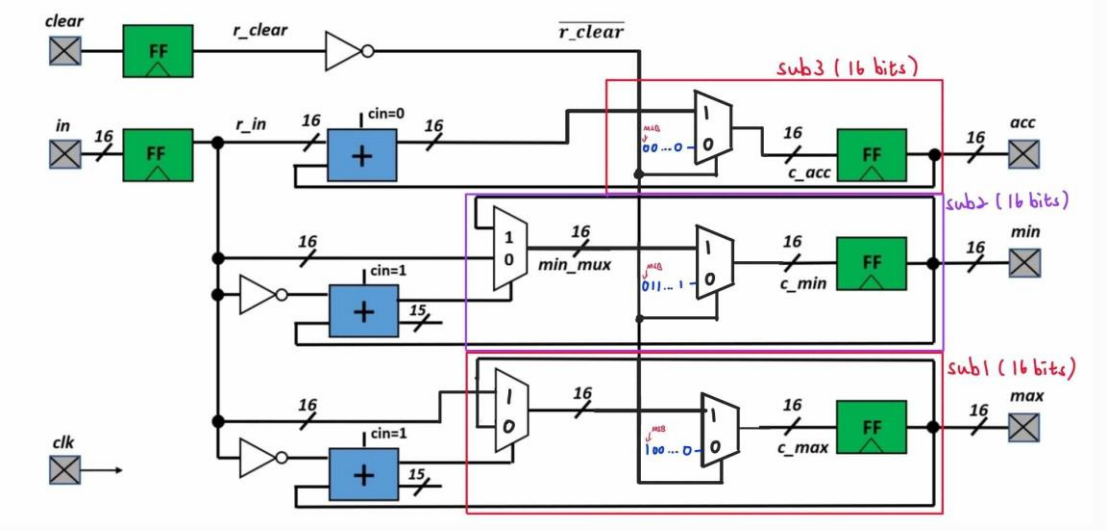
Final Project

TEAM 5

110060021 曾偉博 110060035 黃振寧

Circuit Design

Top cell view



In this design, we first store the 'in' and 'clear' signals in flip-flops during the initial clock cycle. Then, in the second clock cycle, the 16-bit 'in' signal is fed into three separate 16-bit adders. We will consider three different cases, namely 'acc', 'min', and 'max'. For 'acc', the operation performed is an accumulation (acc = acc + in). For 'min' and 'max', subtraction is achieved using the 16-bit adders and inverters (S = min/max + (in' + 1) = min/max - in). Without considering overflow, when S15 (the most significant bit) is '1', it implies min/max - in < 0, indicating min/max < in. Therefore, in the 'max' case, the multiplexer selects 'in' instead of the old 'max', and in the 'min' case, the multiplexer selects the original 'min'. Conversely, when S15 is '0', indicating min/max - in >= 0, min/max >= in, the multiplexer in the 'min' case selects 'in' instead of the old 'min', and in the 'max' case, it retains the original 'max'.

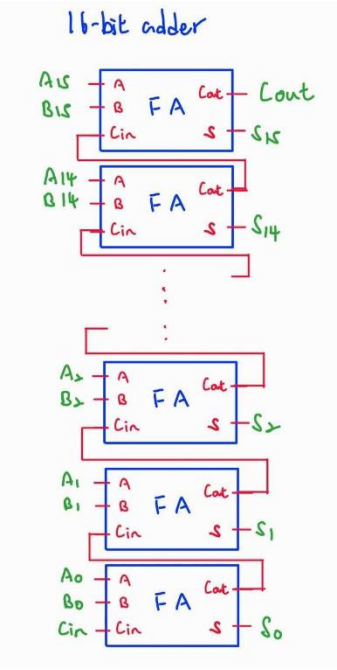
Next, we need to perform initialization. Unlike the reference circuit provided by the teaching assistant, which uses AND/OR or AND blocks with an inverter, we aim to reduce area and circuit complexity by using muxes for initialization. We take the inverted r_clear signal (r_clear', which is generated by passing the 'clear' signal through a flip-flop in the first clock cycle and then through an inverter in the second clock cycle) as the select signal. When r_clear' is '0', the circuit undergoes initialization (conversely, when r_clear' is '1', it selects the result passed from the previous stage). We initialize 'acc' to 0 (16'sb 0000000000000000), 'min' to the maximum value of a 16-bit 2's complement (2^15-1 = 16'sb 0111111111111111), and 'max' to the minimum value of a 16-bit 2's complement (-2^15 = 16'sb 1000000000000000). Finally, the signals pass through another flip-flop stage, yielding the new 'acc', 'min', and 'max' values.

Additionally, to meet the project requirements and enhance the stability of signals within the input flip-flops, we pass both the D and CLK inputs of each flip-flop through two inverters before proceeding with other operations, similar to the first sub-question of Homework 4.

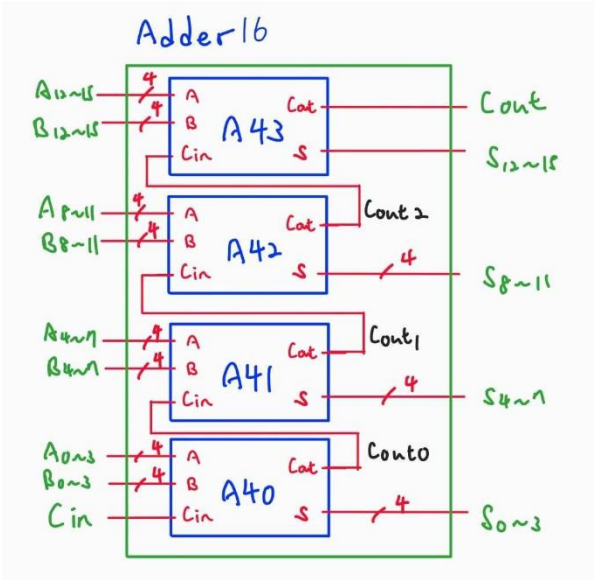
As mentioned above, our goal is to minimize the area and reduce circuit complexity. Additionally, given that the output load capacitance specified in the problem is only 100 femtofarads (fF) per output bit (indicating a small output load without the issue of inadequate drive strength from too small sizes), employing unit size transistors should yield satisfactory results.

Subblocks

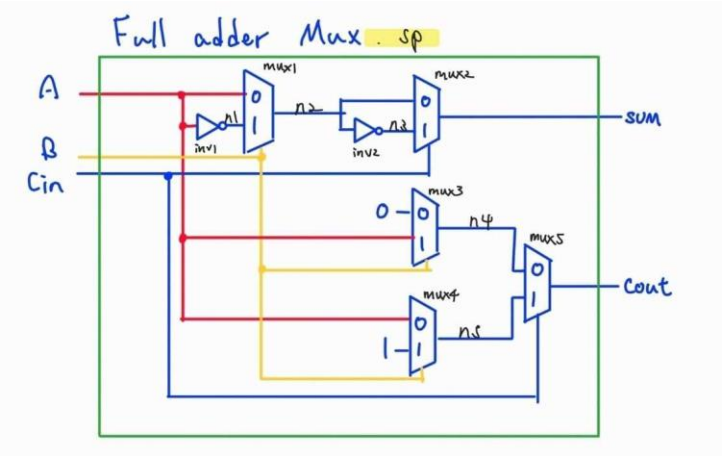
1. 16-bit adder



which is later defined by four 4-bit adder during implementation in netlist.



2. Full adder

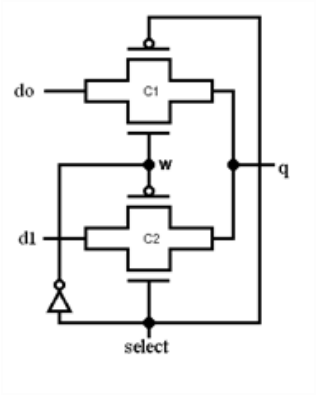


Through the analysis of the truth table, it is evident that the implementation of a full adder can be efficiently achieved using only multiplexers (muxes) and inverters.

In our design, the first stage uses inverters to generate the complement of the input signals. Following this, we employ a series of multiplexers. The first set of multiplexers is configured to generate the preliminary carry and sum signals based on the inverted inputs. The subsequent multiplexers refine these signals by considering all possible input combinations and the corresponding outputs. The final stage multiplexers then consolidate these signals to produce the final carry and sum outputs.

This method not only simplifies the design by reducing the variety of components required but also showcases the functional flexibility of multiplexers. Such a design is beneficial in VLSI applications where **space is at a premium**. Our analysis and simulation results confirm that this mux-inverter full adder design meets all functional requirements, with performance metrics exceeding traditional gate-based designs.

3. 2:1 MUX design (Using transmission gate)



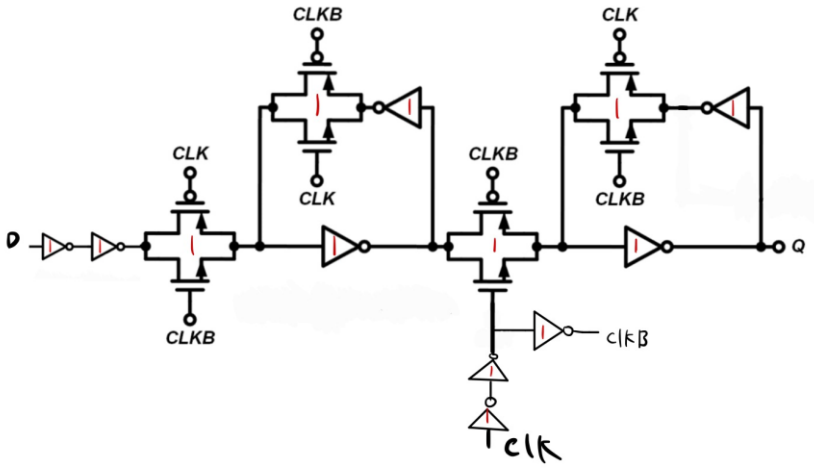
The "select" signal is the control voltage applied to both the n-channel and p-channel MOSFETs in the CMOS transmission gate. It determines whether the gate is conductive or non-conductive:

When the "select" signal is high (logic level '1'), it turns on both transistors connected to D1. The n-channel MOSFET conducts better for a low voltage (logic '0'), and the p-channel MOSFET conducts better for a high voltage (logic '1'). This dual conduction ensures that both logic levels can pass through the gate effectively. When the "select" signal is low (logic level '0'), both MOSFETs connected to D1 are turned off, preventing any signal from passing from the input to the output, vice versa.

Thus, the select signal acts as a binary switch that controls the state of the transmission gate, enabling or disabling the flow of digital signals based on its voltage level.

In situations where space is a limiting factor, this design proves more efficient than employing AND/OR gates for logic implementation. This efficiency in space utilization is a key reason for our decision to adopt this design approach.

4. Flip-flop

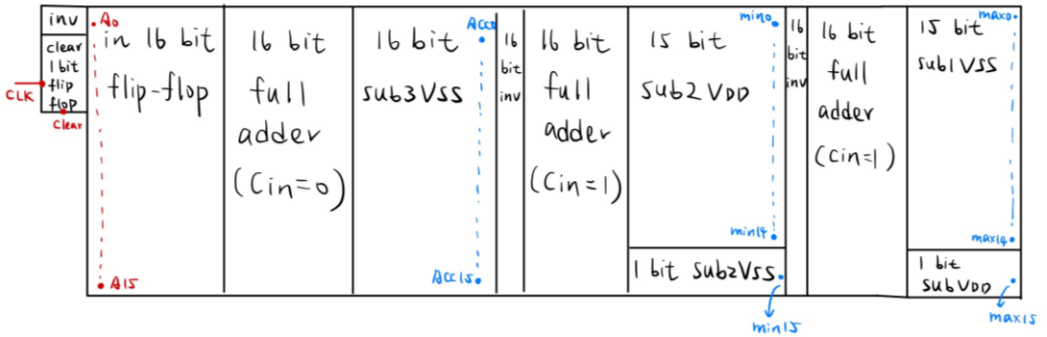


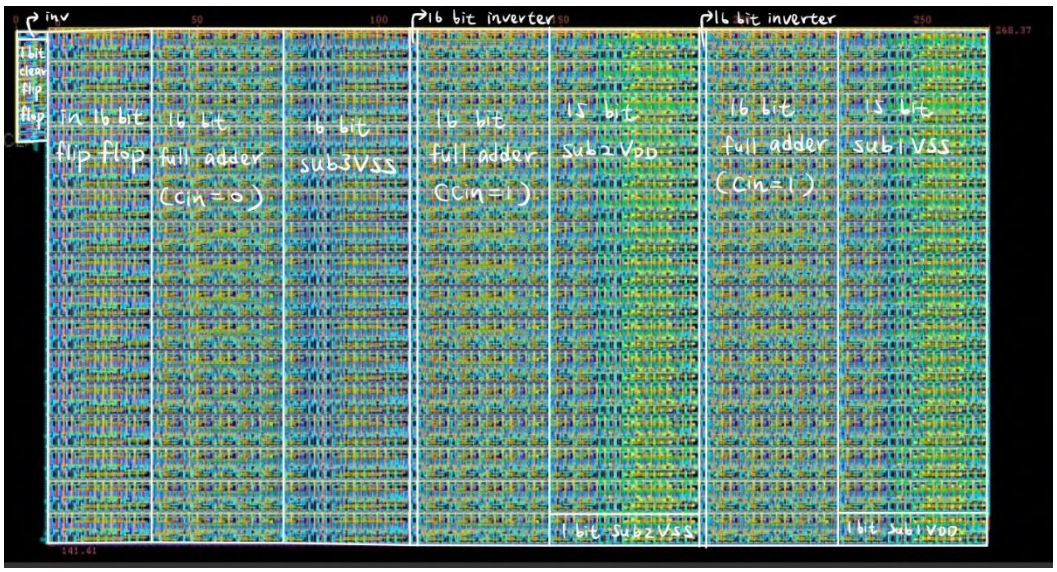
This design adopts a similar approach to that used in the first sub question of Homework 4, with the specification that all inverters and transmission gates are implemented using unit-size components for uniformity and simplicity.

($(\frac{W}{L})_n = \frac{0.5u}{0.18u}, (\frac{W}{L})_p = \frac{1.5u}{0.18u}$) .

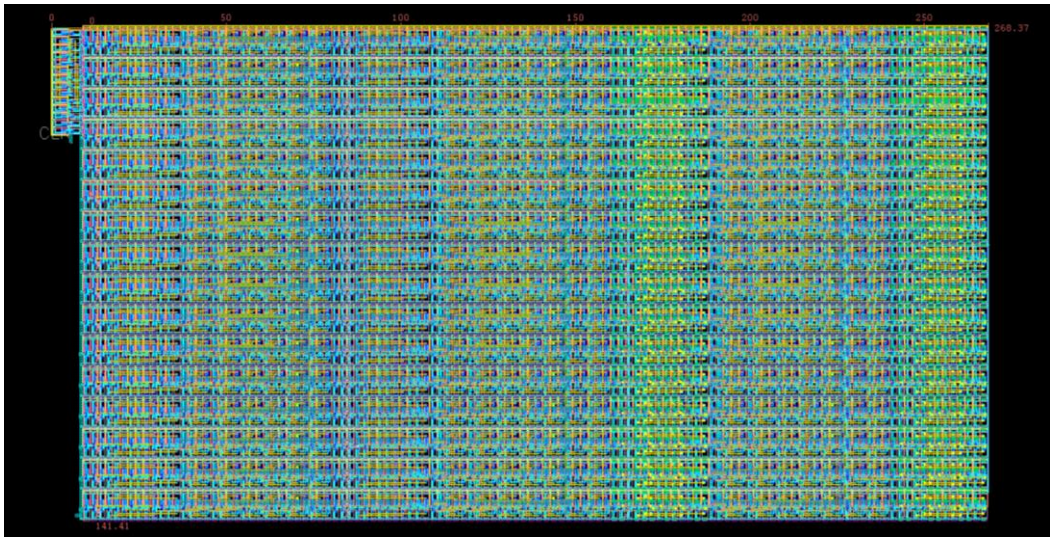
Layout Design

1. Block diagram

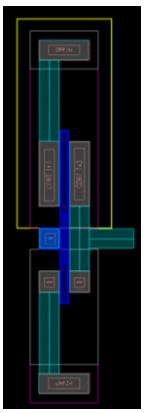
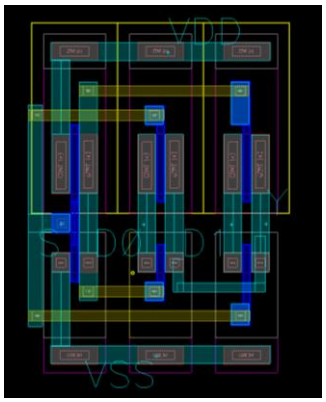




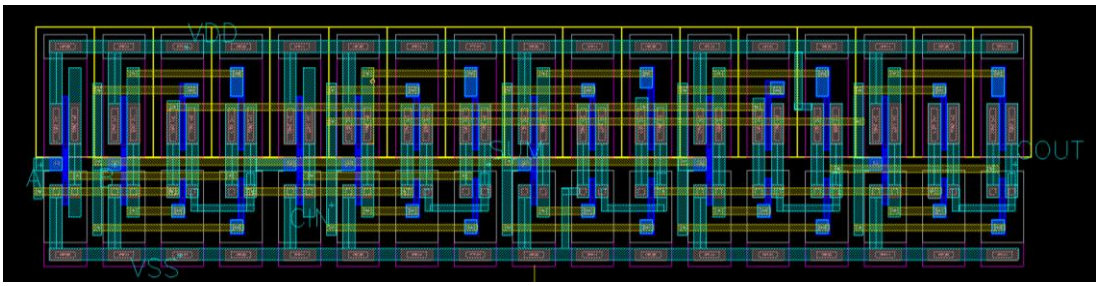
2. TOP (area = 268.37 * 141.41 = 37950.2017 $(um)^2$)



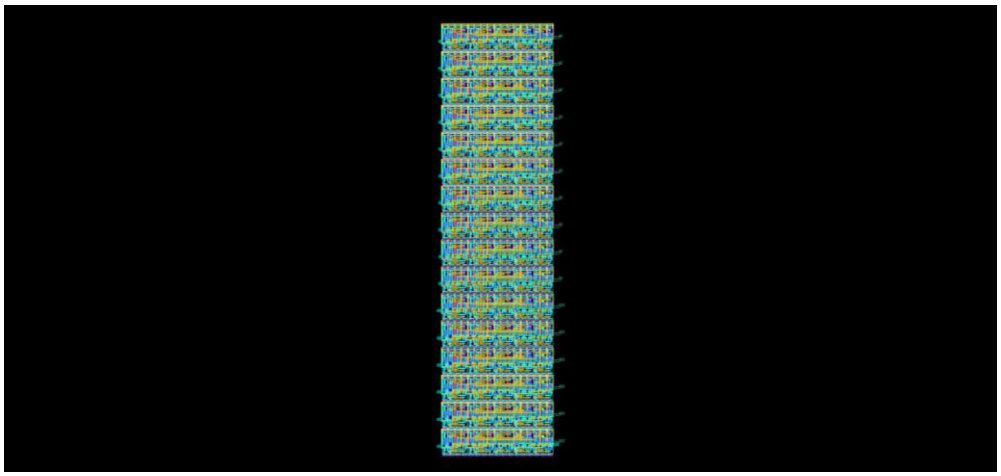
3. Sub blocks

INV	MUX
	

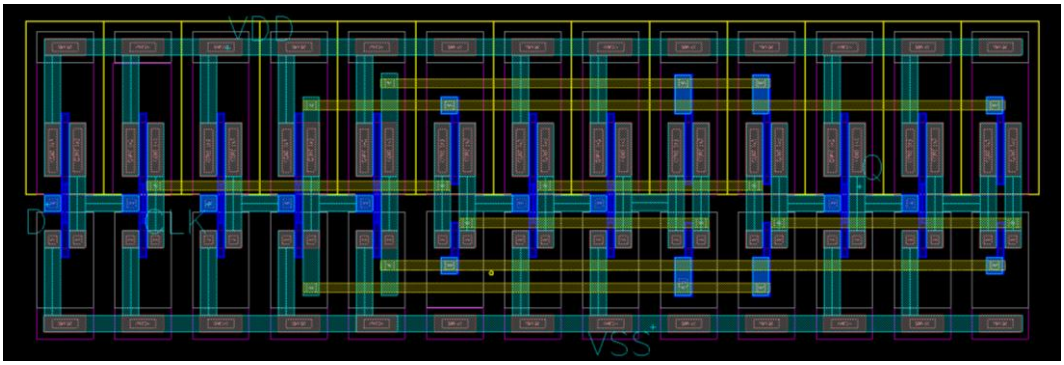
FA



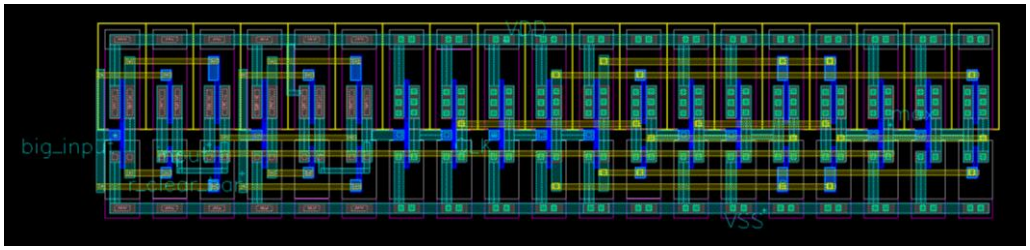
16 bit adder



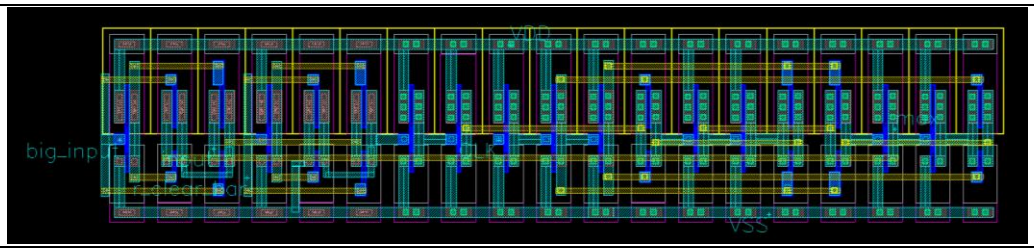
FF



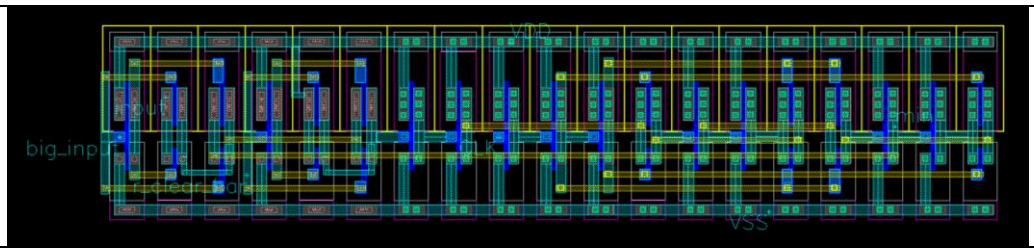
SUB1VDD



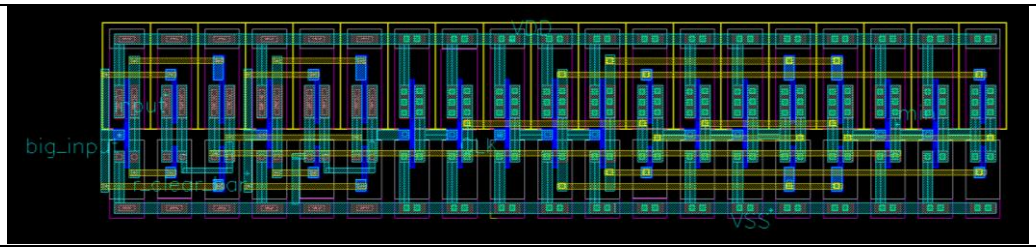
SUB1VSS



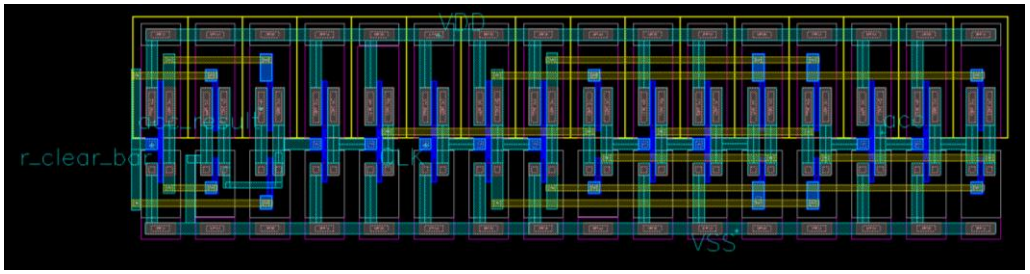
SUB2VDD



SUB2VSS



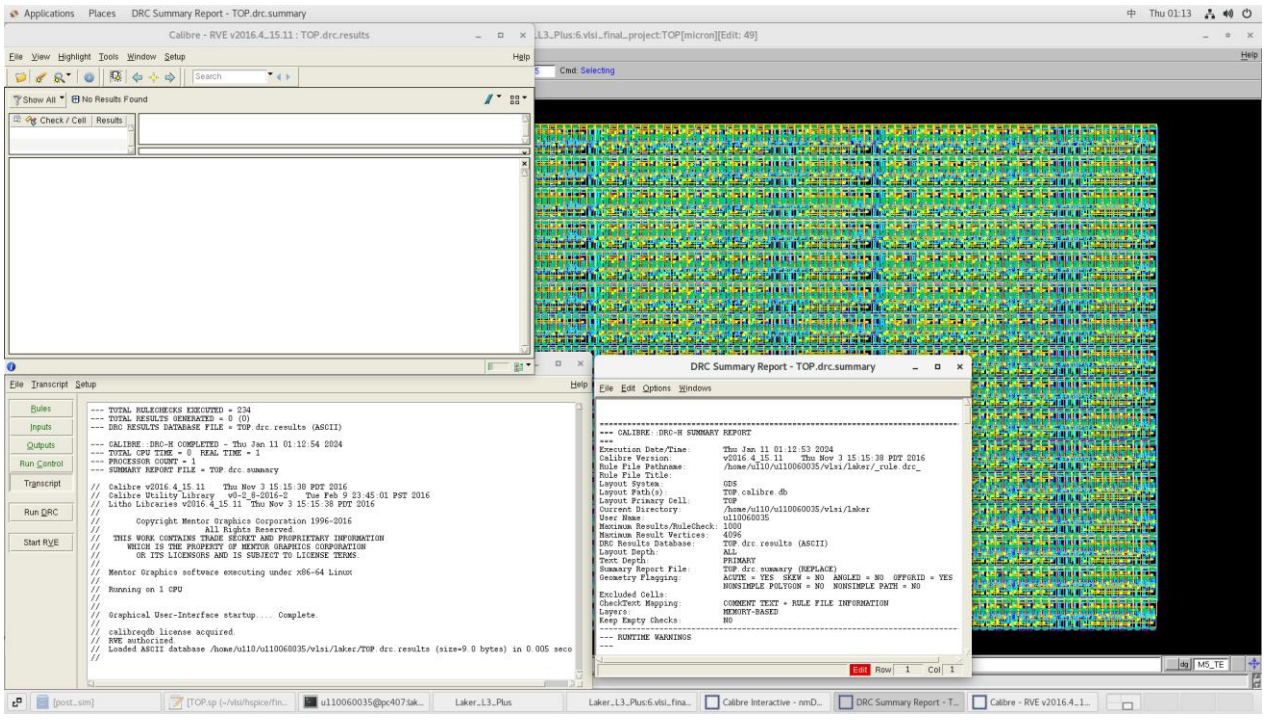
SUB3VSS



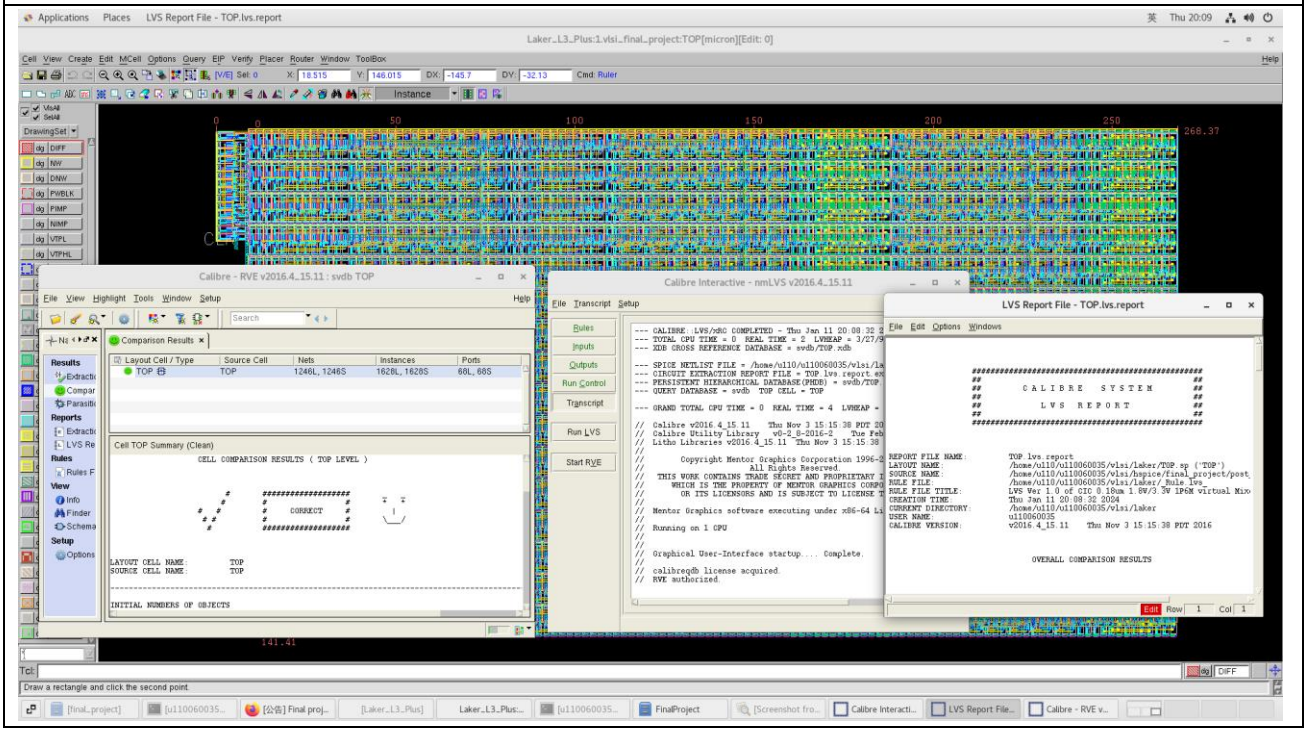
Within our top cell view, we have SUB1, SUB2, and SUB3. The reason for further dividing these into five distinct layouts—SUB1VDD, SUB1VSS, SUB2VDD, SUB2VSS, and SUB3VSS—is due to the differing initialization requirements for the min/max cases, which need to be set to both '1' and '0', as opposed to the acc case, which only needs to be initialized to '0'.

4. Verification

DRC



LVS



5. Layout Consideration

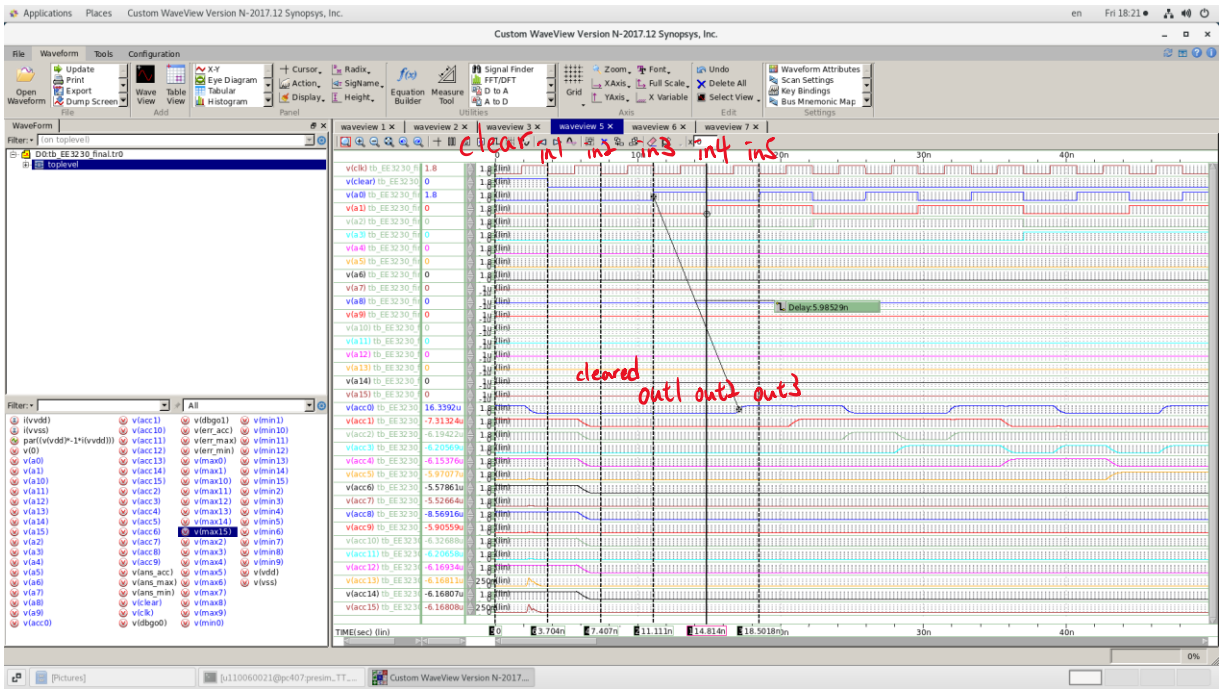
We have meticulously adjusted the heights of each subblock to ensure uniformity, allowing all interconnects to fit without requiring additional area. Each subblock is abutted side-to-side while maintaining a minimum spacing of 0.05 above and below, complying with the minimum separation of 0.25 required between n+ and p+ diffusions. Additionally, given that the 'clear' flip-flop and inverter are single-bit components and cannot form a rectangle with the other 16-bit sections, we've optimized space by rotating them 90 degrees.

Interconnects between different subblocks are as follows: we use metal5 for the CLK, metal1 for VDD and VSS, metal2 for CLEAR, and metal4 for the 'in' signal. The output signals 'acc', 'min', and 'max' are routed back to the 16-bit adder using metal3, which is also employed for the internal 'cin' and 'cout' connections of the 16-bit adder. The sum (S) from the 16-bit adder is connected to SUB3/SUB2/SUB1 using metal3. This layout ensures compact and efficient use of space while maintaining clear signal paths for functionality.

Simulation Result

1. PreSim (TT 25)

   Mode 0

Acc


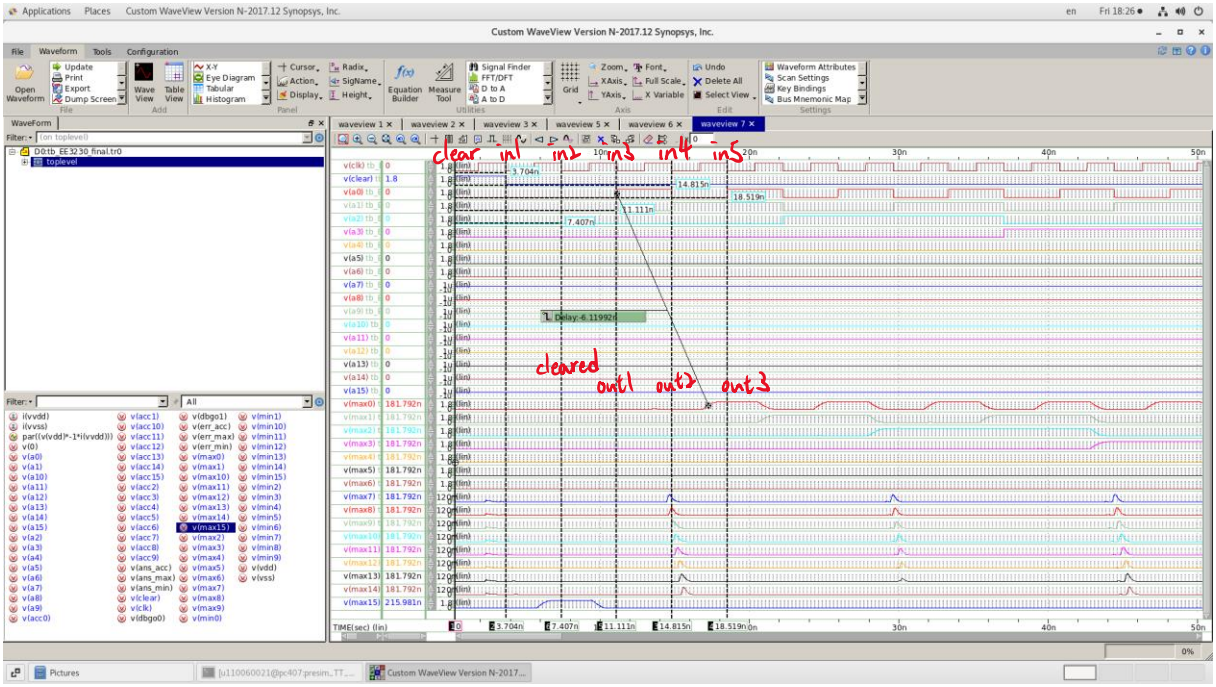
The labels 'clear', 'in1', 'in2', etc., denote distinct input signals. The corresponding outputs 'cleared', 'out1', 'out2', etc., are reflected after a delay of two clock cycles. For instance, at the start of the simulation, the 'clear' signal is activated (logic high), leading to the initialization of the output 'acc0' through 'acc15'. This initialization process completes within two clock cycles, as can be seen at the 'cleared' label, where the output assumes the initial value of 16'sb0 (binary representation of 0 in a 16-bit register).

The maximum operating frequency for our design is 0.27 GHz, corresponding to a clock period of $\frac{1}{2.7GHz}$, which calculates to approximately 370.37 ps. The observed input-to-output delay is 5.98529 ns, which complies with the requirement that the delay should not exceed twice the clock period (2*tclk). It's important to note that the worst-case delay scenario should approach but not exceed 2*tclk. The larger margin observed here is due to this being a sample case rather than the worst-case scenario.

From the onset, the 'clear' signal initializes the 'acc' to 0, with 'acc' being updated after two clock cycles. The 'acc' output signal, as observed, correctly accumulates the input values (a0~a7). For the given input sequence of 0, 0, 1, 2, etc., the accumulation sequence for 'acc' is

0, 0, 1, 3, etc., demonstrating that the design is functioning as intended with sequential input values.

Max

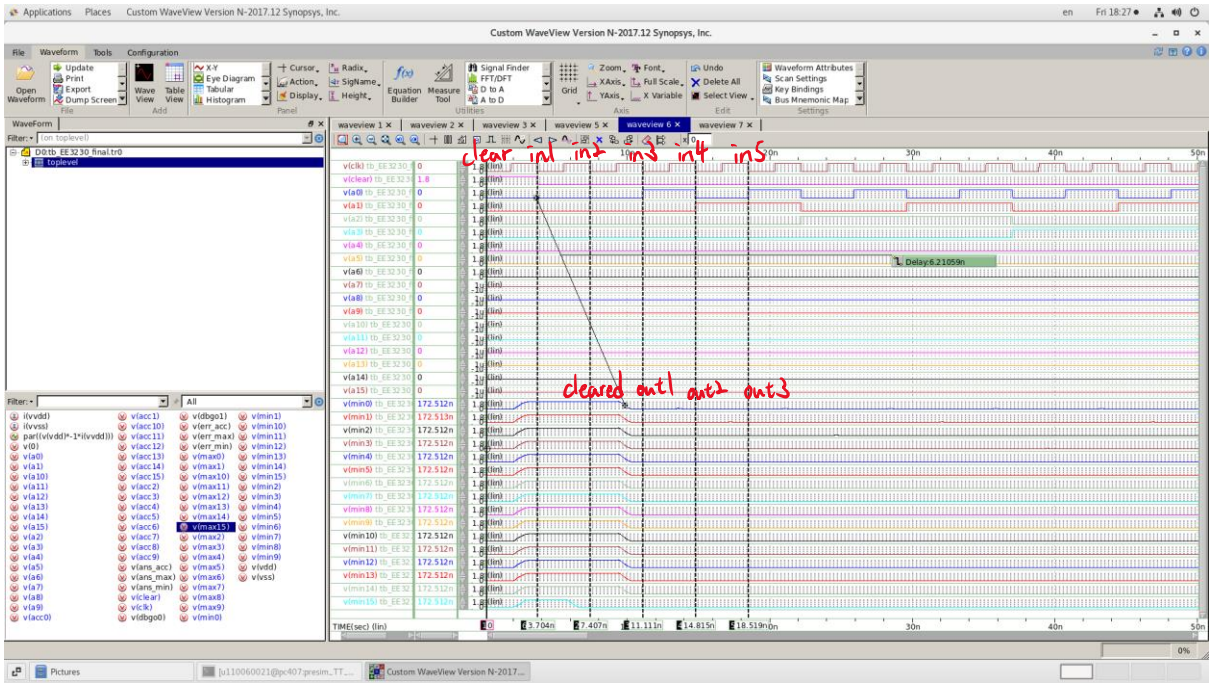


The labels 'clear', 'in1', 'in2', etc., denote distinct input signals. The corresponding outputs 'cleared', 'out1', 'out2', etc., are reflected after a delay of two clock cycles. For instance, at the start of the simulation, the 'clear' signal is activated (logic high), leading to the initialization of the output 'max0' through 'max15'. This initialization process completes within two clock cycles, as can be seen at the 'cleared' label, where the output assumes the initial value of 16'sb1000000000000000 (binary representation of -2^15 in a 16-bit register).

The maximum operating frequency for our design is 0.27 GHz, corresponding to a clock period of $\frac{1}{2.7GHz}$, which calculates to approximately 370.37 ps. The observed input-to-output delay is 6.11992 ns, which complies with the requirement that the delay should not exceed twice the clock period (2*tclk). It's important to note that the worst-case delay scenario should approach but not exceed 2*tclk. The larger margin observed here is due to this being a sample case rather than the worst-case scenario.

From the onset, the 'clear' signal initializes the 'max' to 0, with ‘max’ being updated after two clock cycles. The 'max' output signal, as observed, correctly accumulates the input values (a0~a7). For the given input sequence of 0, 0, 1, 2, etc., the maximum sequence for 'max' is 0, 0, 1, 2, etc., demonstrating that the design is functioning as intended with sequential input values.

Min



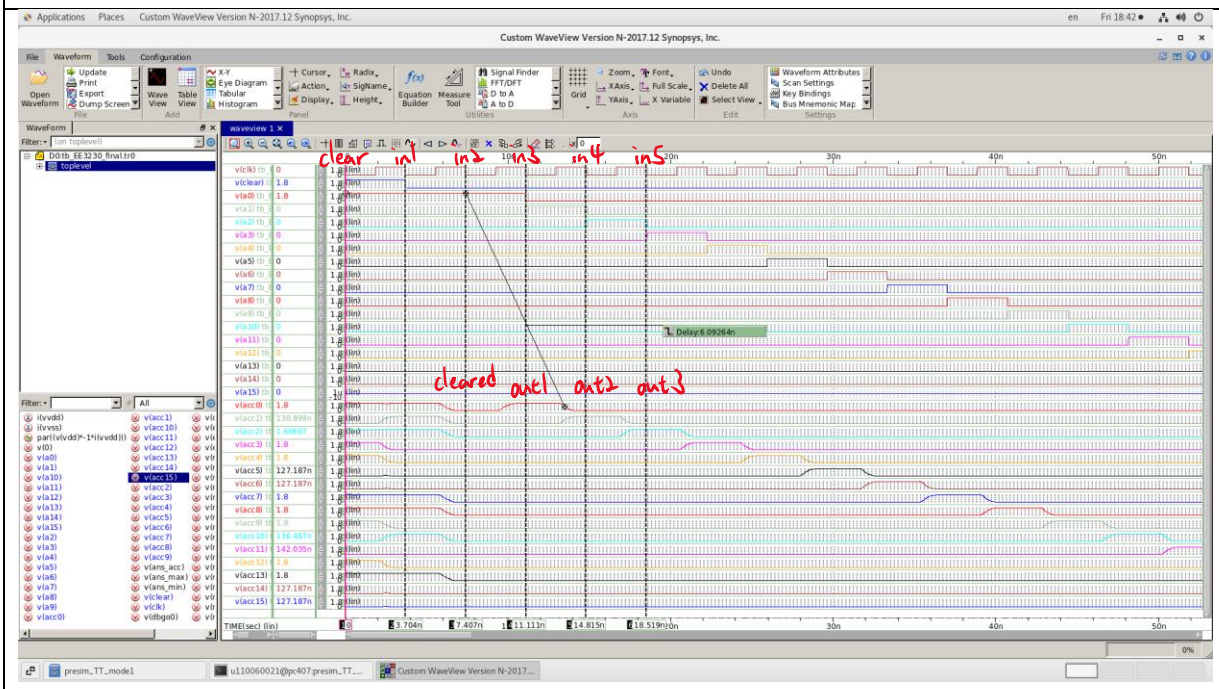
The labels 'clear', 'in1', 'in2', etc., denote distinct input signals. The corresponding outputs 'cleared', 'out1', 'out2', etc., are reflected after a delay of two clock cycles. For instance, at the start of the simulation, the 'clear' signal is activated (logic high), leading to the initialization of the output 'min0' through 'min15'. This initialization process completes within two clock cycles, as can be seen at the 'cleared' label, where the output assumes the initial value of 16'sb0111111111111111 (binary representation of 2^15-1 in a 16-bit register).

The maximum operating frequency for our design is 0.27 GHz, corresponding to a clock period of$\frac{1}{2.7GHz}$, which calculates to approximately 370.37 ps. The observed input-to-output delay is 6.21059 ns, which complies with the requirement that the delay should not exceed twice the clock period (2*tclk). It's important to note that the worst-case delay scenario should approach but not exceed 2*tclk. The larger margin observed here is due to this being a sample case rather than the worst-case scenario.

From the onset, the 'clear' signal initializes the ‘min’ to 0, with 'min' being updated after two clock cycles. The ‘min’ output signal, as observed, correctly accumulates the input values (a0~a7). For the given input sequence of 0, 0, 1, 2, etc., the minimum is always 0, demonstrating that the design is functioning as intended with sequential input values.

Mode 1

Acc



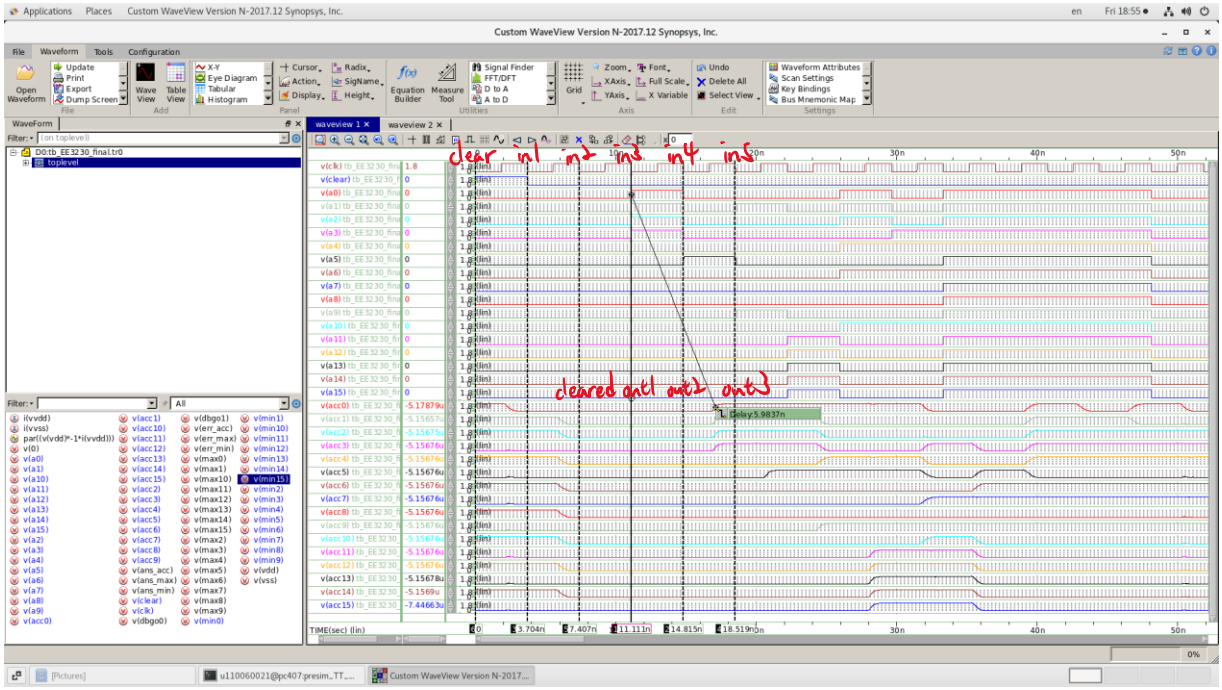
The labels 'clear', 'in1', 'in2', etc., denote distinct input signals. The corresponding outputs 'cleared', 'out1', 'out2', etc., are reflected after a delay of two clock cycles. For instance, at the start of the simulation, the 'clear' signal is activated (logic high), leading to the initialization of the output 'acc0' through 'acc15'. This initialization process completes within two clock cycles, as can be seen at the 'cleared' label, where the output assumes the initial value of 16'sb0 (binary representation of 0 in a 16-bit register).

The maximum operating frequency for our design is 0.27 GHz, corresponding to a clock period of $\frac{1}{2.7GHz}$, which calculates to approximately 370.37 ps. The observed input-to-output delay is 6.09264 ns, which complies with the requirement that the delay should not exceed twice the clock period (2*tclk). It's important to note that the worst-case delay scenario should approach but not exceed 2*tclk. The larger margin observed here is due to this being a sample case rather than the worst-case scenario.

From the onset, the 'clear' signal initializes the 'acc' to 0, with 'acc' being updated after two clock cycles. The 'acc' output signal, as observed, correctly accumulates the input values (a0~a7). For the given input sequence of 1, 1, 2, 4, 8 etc., the accumulation sequence for 'acc' is 1, 2, 4, 8. 16, etc., demonstrating that the design is functioning as intended with sequential input values.

Mode 2

Acc

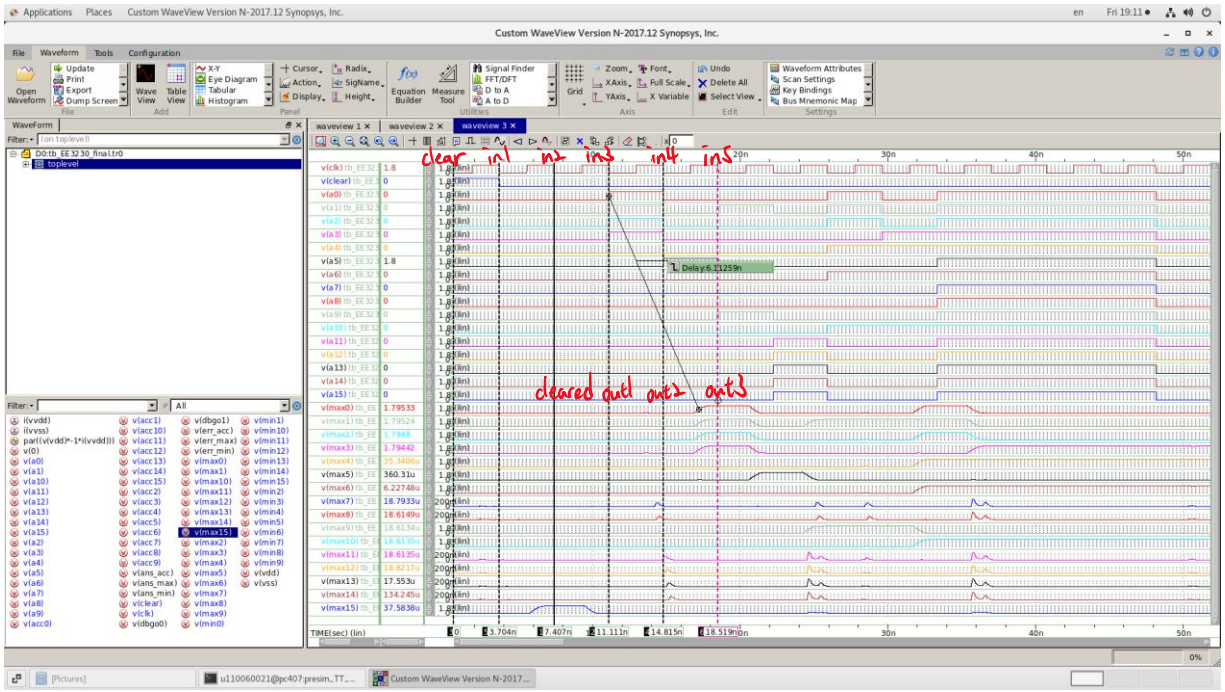


The labels 'clear', 'in1', 'in2', etc., denote distinct input signals. The corresponding outputs 'cleared', 'out1', 'out2', etc., are reflected after a delay of two clock cycles. For instance, at the start of the simulation, the 'clear' signal is activated (logic high), leading to the initialization of the output 'acc0' through 'acc15'. This initialization process completes within two clock cycles, as can be seen at the 'cleared' label, where the output assumes the initial value of 16'sb0 (binary representation of 0 in a 16-bit register).

The maximum operating frequency for our design is 0.27 GHz, corresponding to a clock period of $\frac{1}{2.7GHz}$, which calculates to approximately 370.37 ps. The observed input-to-output delay is 5.9837 ns, which complies with the requirement that the delay should not exceed twice the clock period (2*tclk). It's important to note that the worst-case delay scenario should approach but not exceed 2*tclk. The larger margin observed here is due to this being a sample case rather than the worst-case scenario.

From the onset, the 'clear' signal initializes the 'acc' to 0, with 'acc' being updated after two clock cycles. The 'acc' output signal, as observed, correctly accumulates the input values (a0~a7). For the given input sequence of 0, 0, 15, 32 etc., the accumulation sequence for 'acc' is 0, 0, 15, 47 etc., demonstrating that the design is functioning as intended with sequential input values.

Max

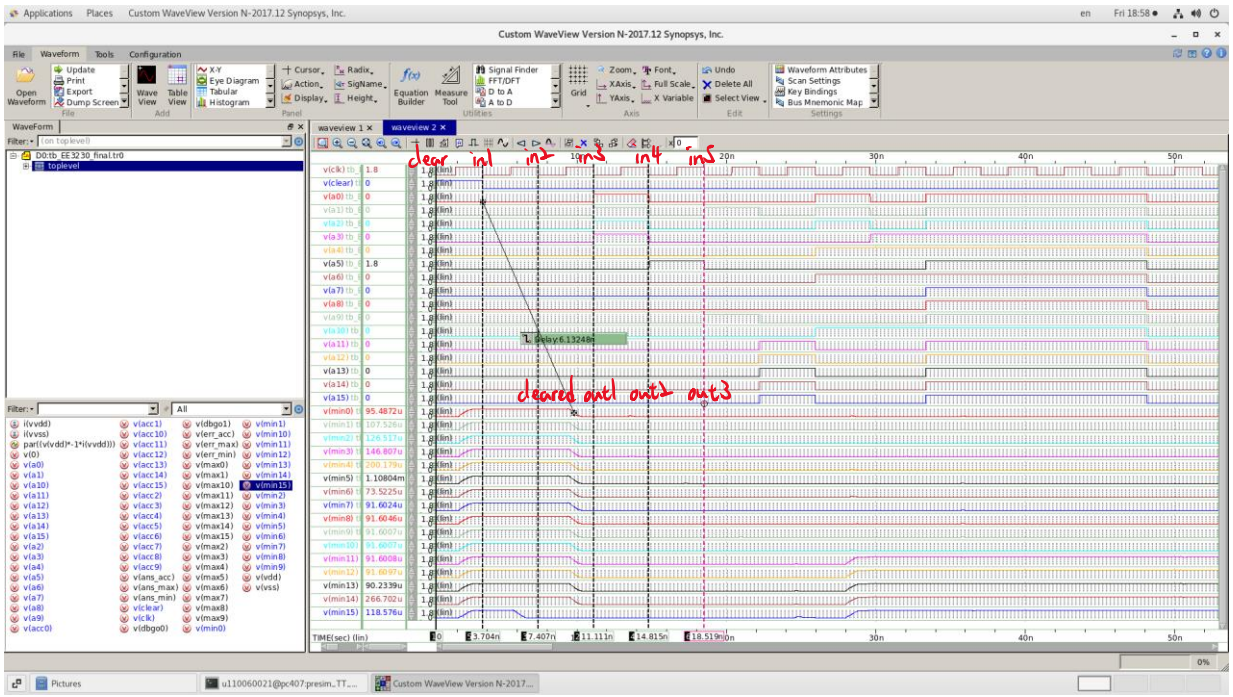


The labels 'clear', 'in1', 'in2', etc., denote distinct input signals. The corresponding outputs 'cleared', 'out1', 'out2', etc., are reflected after a delay of two clock cycles. For instance, at the start of the simulation, the 'clear' signal is activated (logic high), leading to the initialization of the output 'max0' through 'max15'. This initialization process completes within two clock cycles, as can be seen at the 'cleared' label, where the output assumes the initial value of 16'sb1000000000000000 (binary representation of -2^15 in a 16-bit register).

The maximum operating frequency for our design is 0.27 GHz, corresponding to a clock period of$\frac{1}{2.7GHz}$, which calculates to approximately 370.37 ps. The observed input-to-output delay is 6.11259 ns, which complies with the requirement that the delay should not exceed twice the clock period (2*tclk). It's important to note that the worst-case delay scenario should approach but not exceed 2*tclk. The larger margin observed here is due to this being a sample case rather than the worst-case scenario.

From the onset, the 'clear' signal initializes the 'max' to 0, with ‘max’ being updated after two clock cycles. The 'max' output signal, as observed, correctly accumulates the input values (a0~a7). For the given input sequence of 0, 0, 15, 32, etc., the maximum sequence for 'max' is 0, 0, 15, 32, etc., demonstrating that the design is functioning as intended with sequential input values.

Min



The labels 'clear', 'in1', 'in2', etc., denote distinct input signals. The corresponding outputs 'cleared', 'out1', 'out2', etc., are reflected after a delay of two clock cycles. For instance, at the start of the simulation, the 'clear' signal is activated (logic high), leading to the initialization of the output 'min0' through 'min15'. This initialization process completes within two clock cycles, as can be seen at the 'cleared' label, where the output assumes the initial value of 16'sb0111111111111111 (binary representation of 2^15-1 in a 16-bit register).

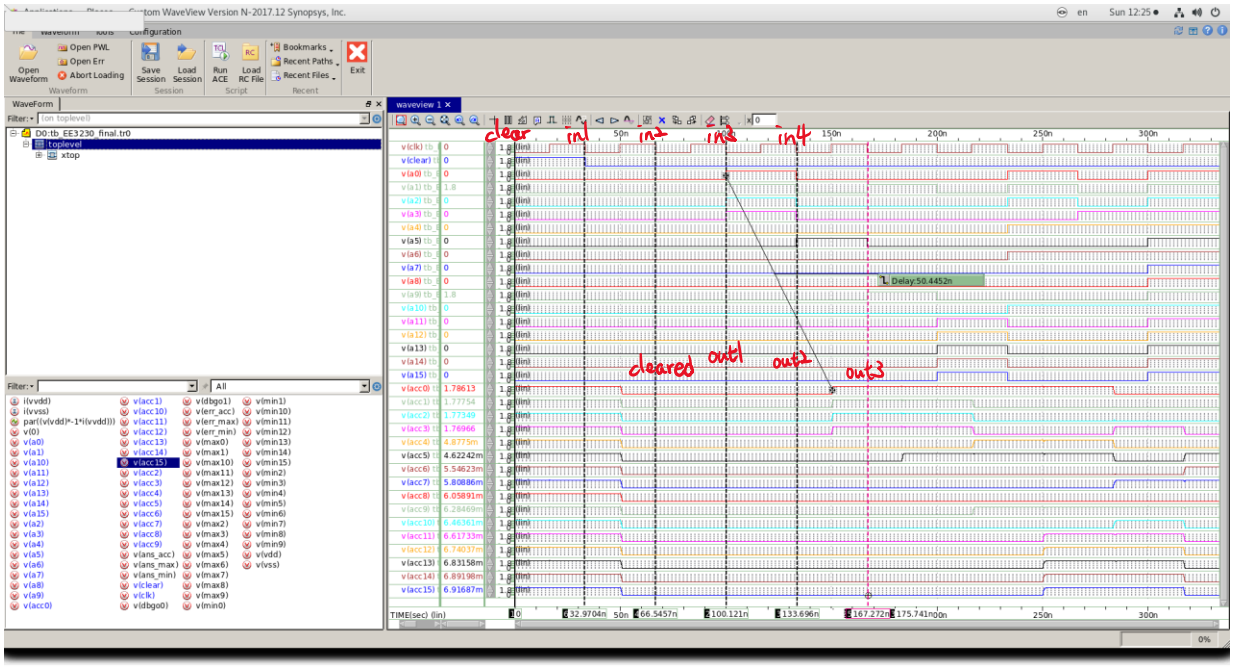
The maximum operating frequency for our design is 2.7 GHz, corresponding to a clock period of$\frac{1}{2.7GHz}$, which calculates to approximately 370.37 ps. The observed input-to-output delay is 6.13248 ns, which complies with the requirement that the delay should not exceed twice the clock period (2*tclk). It's important to note that the worst-case delay scenario should approach but not exceed 2*tclk. The larger margin observed here is due to this being a sample case rather than the worst-case scenario.

From the onset, the 'clear' signal initializes the ‘min’ to 0, with 'min' being updated after two clock cycles. The ‘min’ output signal, as observed, correctly accumulates the input values (a0~a7). For the given input sequence of 0, 0, 15, 32, etc., the minimum is always 0, demonstrating that the design is functioning as intended with sequential input values.

2. PostSim (FF)

Mode 2

Acc

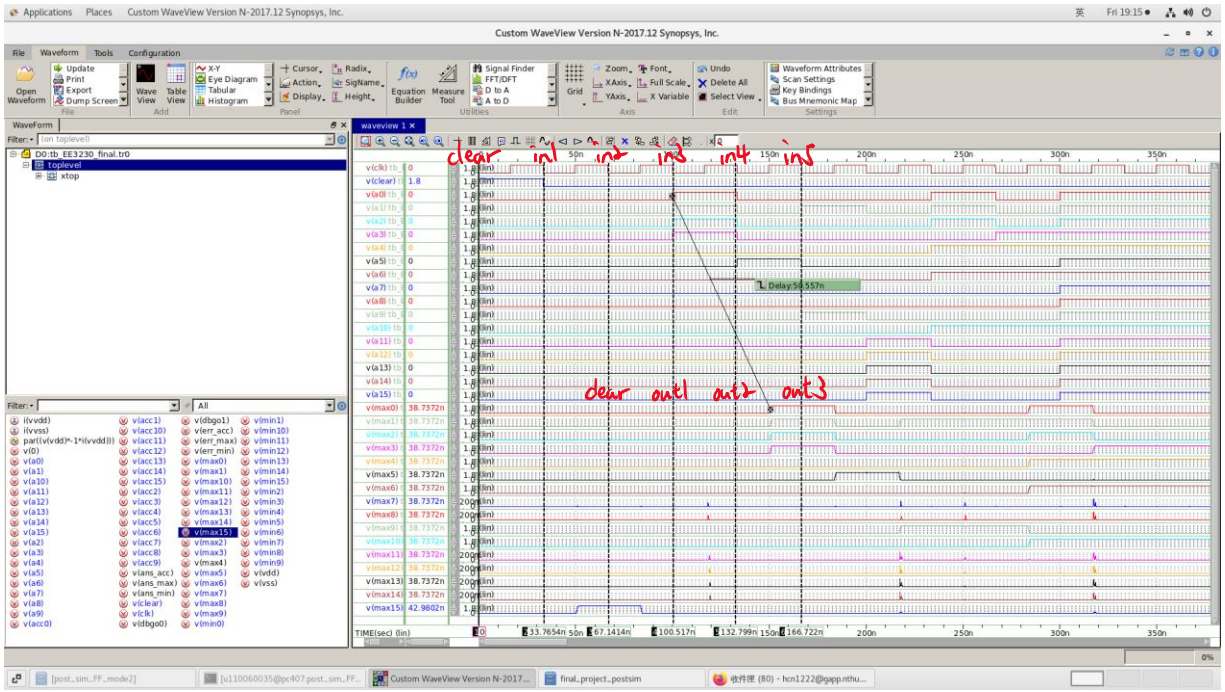


The labels 'clear', 'in1', 'in2', etc., denote distinct input signals. The corresponding outputs 'cleared', 'out1', 'out2', etc., are reflected after a delay of two clock cycles. For instance, at the start of the simulation, the 'clear' signal is activated (logic high), leading to the initialization of the output 'acc0' through 'acc15'. This initialization process completes within two clock cycles, as can be seen at the 'cleared' label, where the output assumes the initial value of 16'sb0 (binary representation of 0 in a 16-bit register).

The operating frequency for this simulation is 0.03 GHz (larger than 10MHz = 0.01 GHz), corresponding to a clock period of $\frac{1}{0.03GHz}$, which calculates to approximately 33.333 ns. The observed input-to-output delay is 50.4452 ns, which complies with the requirement that the delay should not exceed twice the clock period (2*tclk). It's important to note that the worst-case delay scenario should approach but not exceed 2*tclk. The larger margin observed here is due to this being a sample case rather than the worst-case scenario.

From the onset, the 'clear' signal initializes the 'acc' to 0, with 'acc' being updated after two clock cycles. The 'acc' output signal, as observed, correctly accumulates the input values (a0~a7). For the given input sequence of 0, 0, 15, 32 etc., the accumulation sequence for 'acc' is 0, 0, 15, 47 etc., demonstrating that the design is functioning as intended with sequential input values.

Max

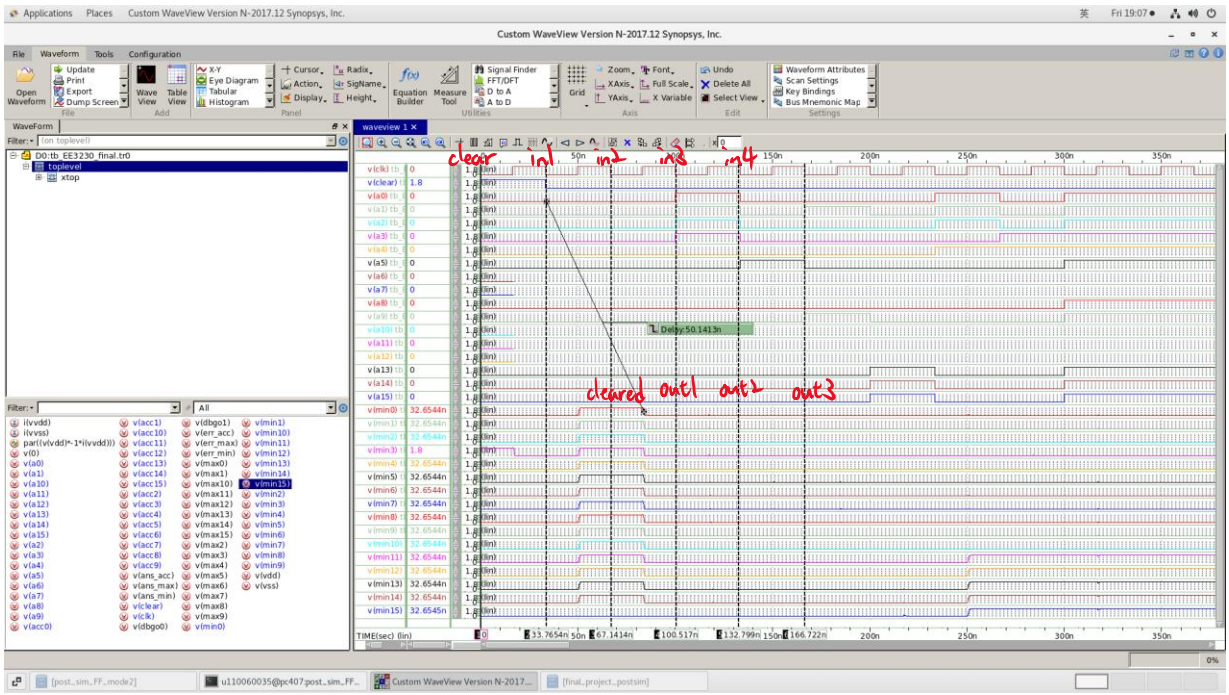


The labels 'clear', 'in1', 'in2', etc., denote distinct input signals. The corresponding outputs 'cleared', 'out1', 'out2', etc., are reflected after a delay of two clock cycles. For instance, at the start of the simulation, the 'clear' signal is activated (logic high), leading to the initialization of the output 'max0' through 'max15'. This initialization process completes within two clock cycles, as can be seen at the 'cleared' label, where the output assumes the initial value of 16'sb1000000000000000 (binary representation of -2^15 in a 16-bit register).

The operating frequency for this simulation is 0.03 GHz (larger than 10MHz = 0.01 GHz), corresponding to a clock period of $\frac{1}{0.03GHz}$, which calculates to approximately 33.333 ns. The observed input-to-output delay is 50.557 ns, which complies with the requirement that the delay should not exceed twice the clock period (2*tclk). It's important to note that the worst-case delay scenario should approach but not exceed 2*tclk. The larger margin observed here is due to this being a sample case rather than the worst-case scenario.

From the onset, the 'clear' signal initializes the 'max' to 0, with ‘max’ being updated after two clock cycles. The 'max' output signal, as observed, correctly accumulates the input values (a0~a7). For the given input sequence of 0, 0, 15, 32, etc., the maximum sequence for 'max' is 0, 0, 15, 32, etc., demonstrating that the design is functioning as intended with sequential input values.

Min



The labels 'clear', 'in1', 'in2', etc., denote distinct input signals. The corresponding outputs 'cleared', 'out1', 'out2', etc., are reflected after a delay of two clock cycles. For instance, at the start of the simulation, the 'clear' signal is activated (logic high), leading to the initialization of the output 'min0' through 'min15'. This initialization process completes within two clock cycles, as can be seen at the 'cleared' label, where the output assumes the initial value of 16'sb0111111111111111 (binary representation of 2^15-1 in a 16-bit register).

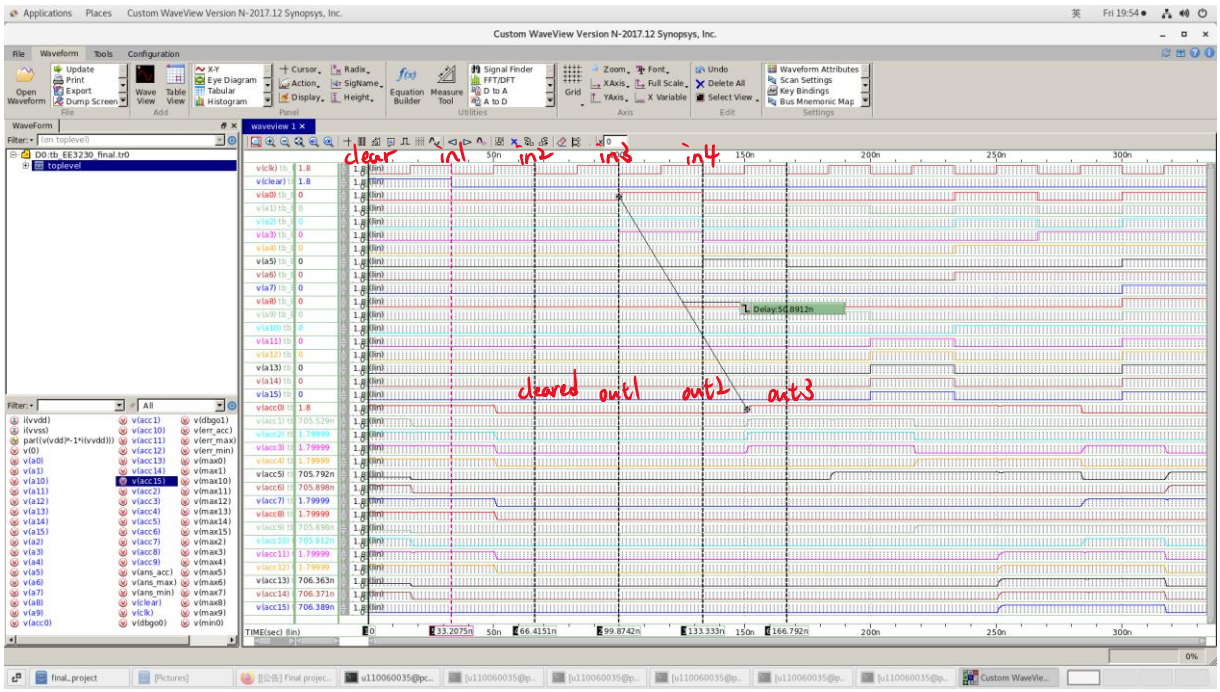
The operating frequency for this simulation is 0.03 GHz (larger than 10MHz = 0.01 GHz), corresponding to a clock period of $\frac{1}{0.03GHz}$, which calculates to approximately 33.333 ns. The observed input-to-output delay is 50.1413 ns, which complies with the requirement that the delay should not exceed twice the clock period (2*tclk). It's important to note that the worst-case delay scenario should approach but not exceed 2*tclk. The larger margin observed here is due to this being a sample case rather than the worst-case scenario.

From the onset, the 'clear' signal initializes the ‘min’ to 0, with 'min' being updated after two clock cycles. The ‘min’ output signal, as observed, correctly accumulates the input values (a0~a7). For the given input sequence of 0, 0, 15, 32, etc., the minimum is always 0, demonstrating that the design is functioning as intended with sequential input values.

3. PostSim (FS)

Mode 2

Acc

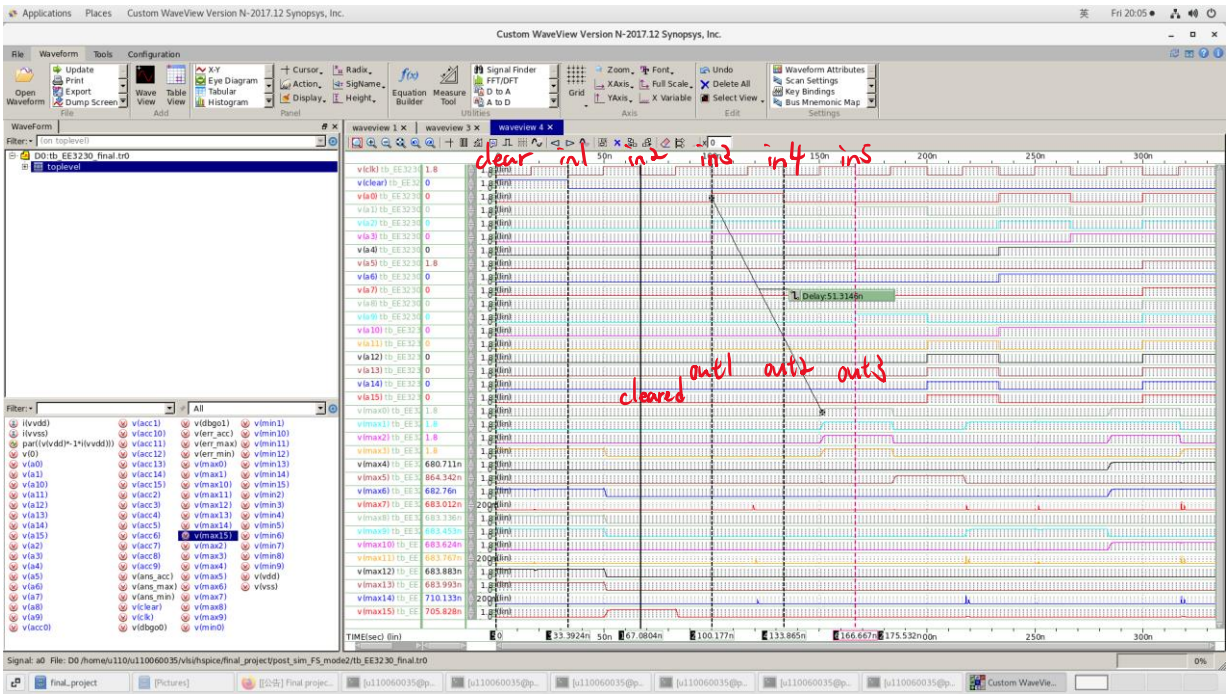


The labels 'clear', 'in1', 'in2', etc., denote distinct input signals. The corresponding outputs 'cleared', 'out1', 'out2', etc., are reflected after a delay of two clock cycles. For instance, at the start of the simulation, the 'clear' signal is activated (logic high), leading to the initialization of the output 'acc0' through 'acc15'. This initialization process completes within two clock cycles, as can be seen at the 'cleared' label, where the output assumes the initial value of 16'sb0 (binary representation of 0 in a 16-bit register).

The operating frequency for this simulation is 0.03 GHz (larger than 10MHz = 0.01 GHz), corresponding to a clock period of $\frac{1}{0.03GHz}$, which calculates to approximately 33.333 ns. The observed input-to-output delay is 50.8912 ns, which complies with the requirement that the delay should not exceed twice the clock period (2*tclk). It's important to note that the worst-case delay scenario should approach but not exceed 2*tclk. The larger margin observed here is due to this being a sample case rather than the worst-case scenario.

From the onset, the 'clear' signal initializes the 'acc' to 0, with 'acc' being updated after two clock cycles. The 'acc' output signal, as observed, correctly accumulates the input values (a0~a7). For the given input sequence of 0, 0, 15, 32 etc., the accumulation sequence for 'acc' is 0, 0, 15, 47 etc., demonstrating that the design is functioning as intended with sequential input values.

Max

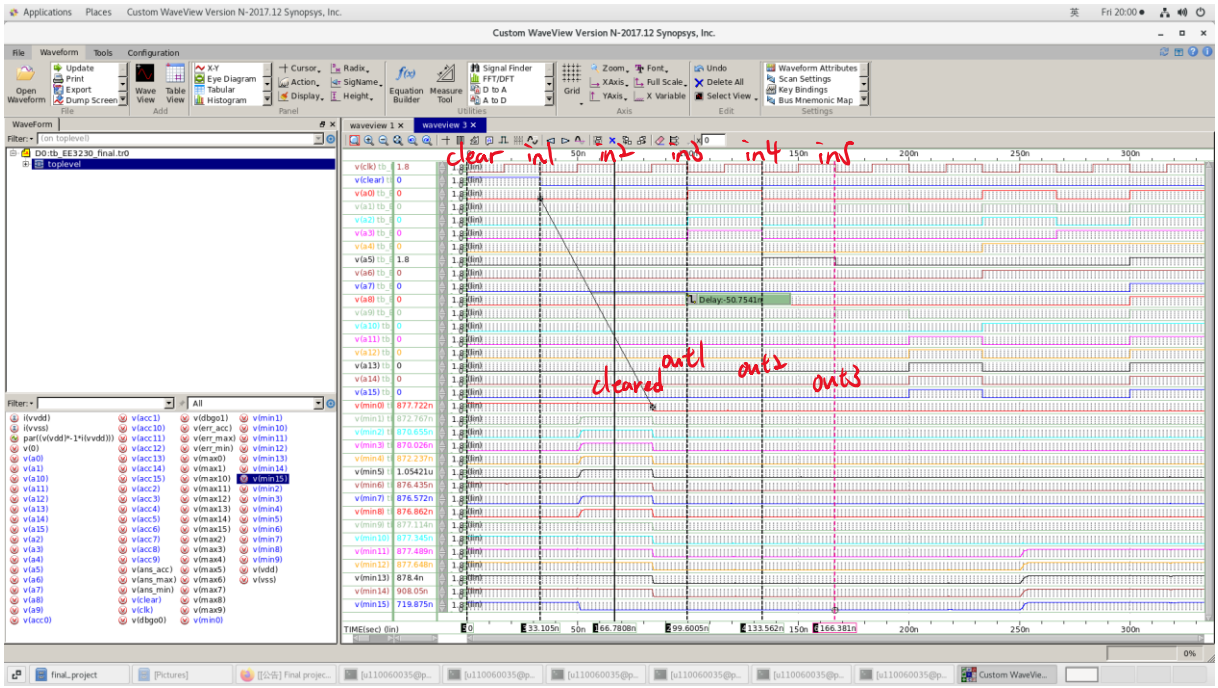


The labels 'clear', 'in1', 'in2', etc., denote distinct input signals. The corresponding outputs 'cleared', 'out1', 'out2', etc., are reflected after a delay of two clock cycles. For instance, at the start of the simulation, the 'clear' signal is activated (logic high), leading to the initialization of the output 'max0' through 'max15'. This initialization process completes within two clock cycles, as can be seen at the 'cleared' label, where the output assumes the initial value of 16'sb1000000000000000 (binary representation of -2^15 in a 16-bit register).

The operating frequency for this simulation is 0.03 GHz (larger than 10MHz = 0.01 GHz), corresponding to a clock period of $\frac{1}{0.03GHz}$, which calculates to approximately 33.333 ns. The observed input-to-output delay is 51.3146 ns, which complies with the requirement that the delay should not exceed twice the clock period (2*tclk). It's important to note that the worst-case delay scenario should approach but not exceed 2*tclk. The larger margin observed here is due to this being a sample case rather than the worst-case scenario.

From the onset, the 'clear' signal initializes the 'max' to 0, with ‘max’ being updated after two clock cycles. The 'max' output signal, as observed, correctly accumulates the input values (a0~a7). For the given input sequence of 0, 0, 15, 32, etc., the maximum sequence for 'max' is 0, 0, 15, 32, etc., demonstrating that the design is functioning as intended with sequential input values.

Min



The labels 'clear', 'in1', 'in2', etc., denote distinct input signals. The corresponding outputs 'cleared', 'out1', 'out2', etc., are reflected after a delay of two clock cycles. For instance, at the start of the simulation, the 'clear' signal is activated (logic high), leading to the initialization of the output 'min0' through 'min15'. This initialization process completes within two clock cycles, as can be seen at the 'cleared' label, where the output assumes the initial value of 16'sb0111111111111111 (binary representation of 2^15-1 in a 16-bit register).

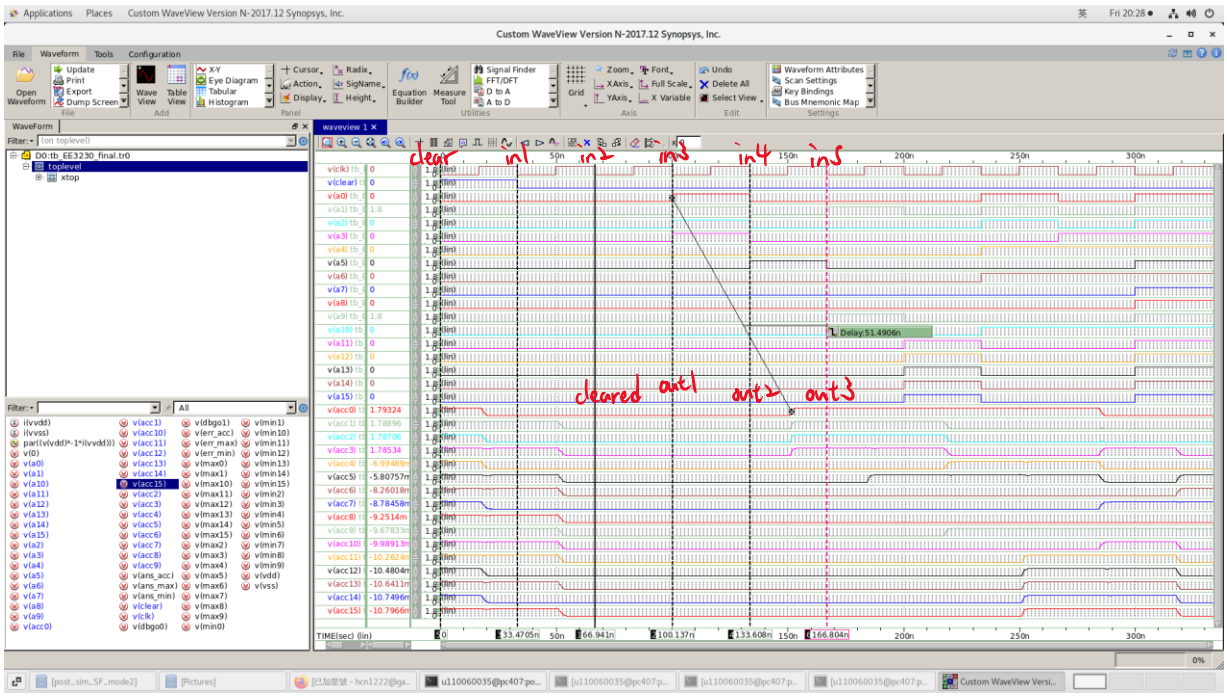
The operating frequency for this simulation is 0.03 GHz (larger than 10MHz = 0.01 GHz), corresponding to a clock period of $\frac{1}{0.03GHz}$, which calculates to approximately 33.333 ns. The observed input-to-output delay is 50.7541 ns, which complies with the requirement that the delay should not exceed twice the clock period (2*tclk). It's important to note that the worst-case delay scenario should approach but not exceed 2*tclk. The larger margin observed here is due to this being a sample case rather than the worst-case scenario.

From the onset, the 'clear' signal initializes the ‘min’ to 0, with 'min' being updated after two clock cycles. The ‘min’ output signal, as observed, correctly accumulates the input values (a0~a7). For the given input sequence of 0, 0, 15, 32, etc., the minimum is always 0, demonstrating that the design is functioning as intended with sequential input values.

4. PostSim (SS)

Mode 2

Acc

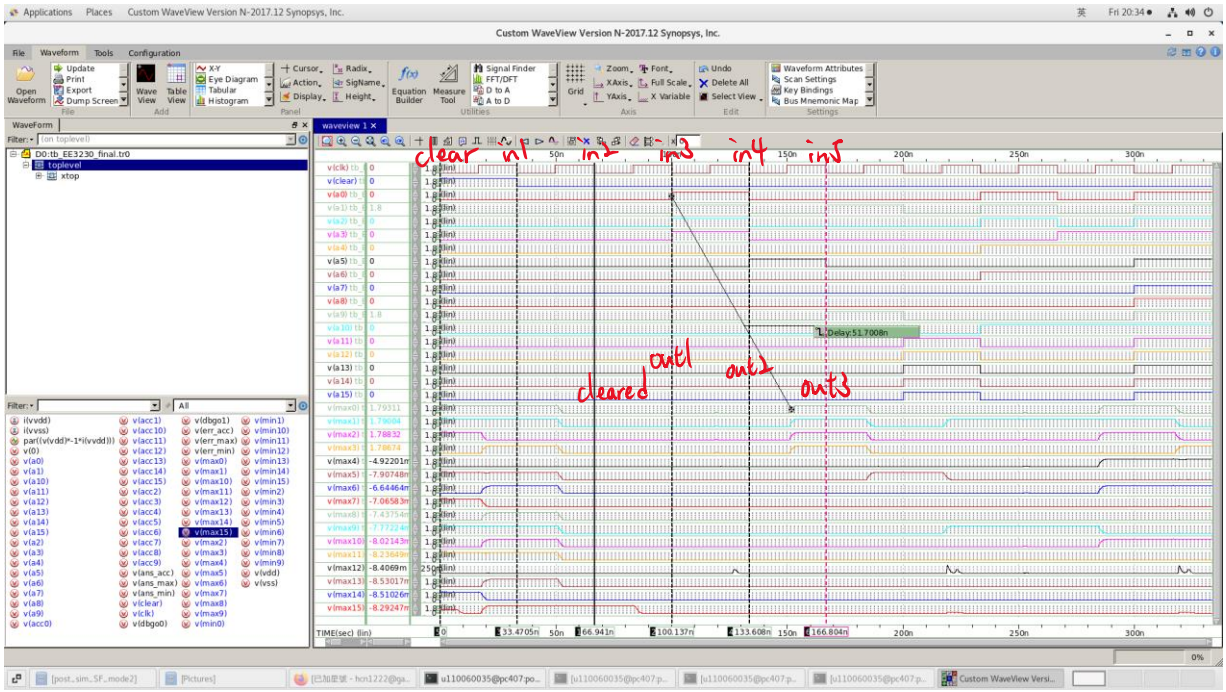


The labels 'clear', 'in1', 'in2', etc., denote distinct input signals. The corresponding outputs 'cleared', 'out1', 'out2', etc., are reflected after a delay of two clock cycles. For instance, at the start of the simulation, the 'clear' signal is activated (logic high), leading to the initialization of the output 'acc0' through 'acc15'. This initialization process completes within two clock cycles, as can be seen at the 'cleared' label, where the output assumes the initial value of 16'sb0 (binary representation of 0 in a 16-bit register).

The operating frequency for this simulation is 0.03 GHz (larger than 10MHz = 0.01 GHz), corresponding to a clock period of $\frac{1}{0.03GHz}$, which calculates to approximately 33.333 ns. The observed input-to-output delay is 51.4906 ns, which complies with the requirement that the delay should not exceed twice the clock period (2*tclk). It's important to note that the worst-case delay scenario should approach but not exceed 2*tclk. The larger margin observed here is due to this being a sample case rather than the worst-case scenario.

From the onset, the 'clear' signal initializes the 'acc' to 0, with 'acc' being updated after two clock cycles. The 'acc' output signal, as observed, correctly accumulates the input values (a0~a7). For the given input sequence of 0, 0, 15, 32 etc., the accumulation sequence for 'acc' is 0, 0, 15, 47 etc., demonstrating that the design is functioning as intended with sequential input values.

Max

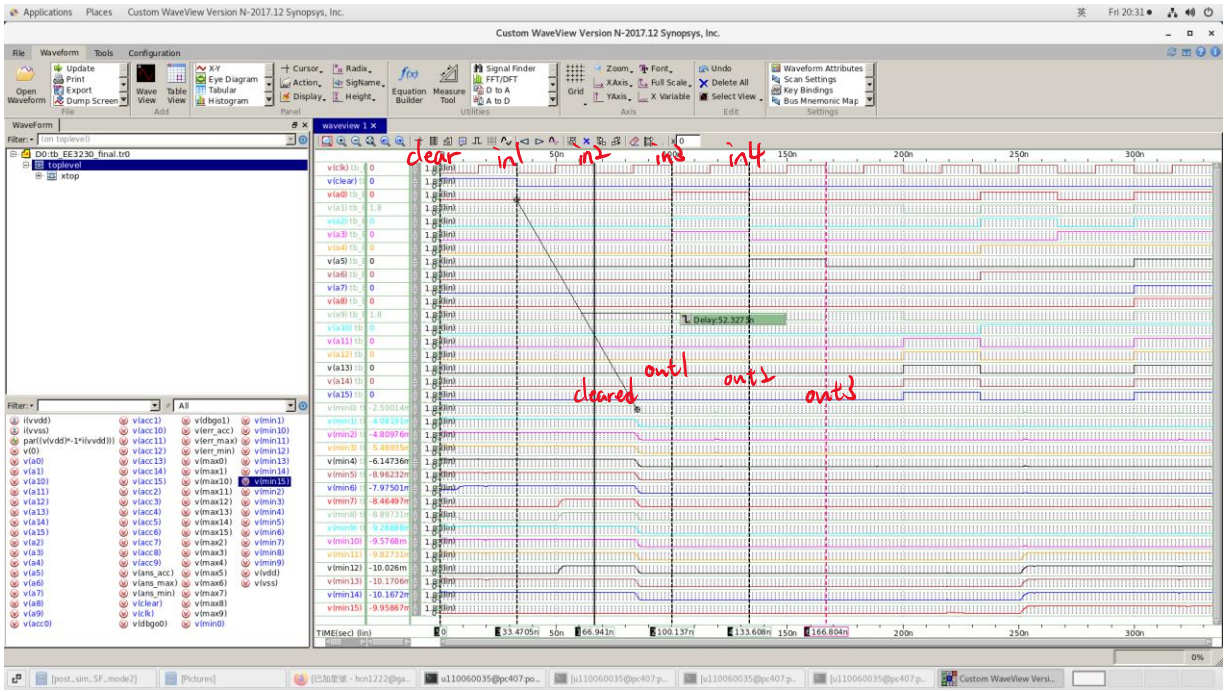


The labels 'clear', 'in1', 'in2', etc., denote distinct input signals. The corresponding outputs 'cleared', 'out1', 'out2', etc., are reflected after a delay of two clock cycles. For instance, at the start of the simulation, the 'clear' signal is activated (logic high), leading to the initialization of the output 'max0' through 'max15'. This initialization process completes within two clock cycles, as can be seen at the 'cleared' label, where the output assumes the initial value of 16'sb1000000000000000 (binary representation of -2^15 in a 16-bit register).

The operating frequency for this simulation is 0.03 GHz (larger than 10MHz = 0.01 GHz), corresponding to a clock period of $\frac{1}{0.03GHz}$, which calculates to approximately 33.333 ns. The observed input-to-output delay is 51.7008 ns, which complies with the requirement that the delay should not exceed twice the clock period (2*tclk). It's important to note that the worst-case delay scenario should approach but not exceed 2*tclk. The larger margin observed here is due to this being a sample case rather than the worst-case scenario.

From the onset, the 'clear' signal initializes the 'max' to 0, with ‘max’ being updated after two clock cycles. The 'max' output signal, as observed, correctly accumulates the input values (a0~a7). For the given input sequence of 0, 0, 15, 32, etc., the maximum sequence for 'max' is 0, 0, 15, 32, etc., demonstrating that the design is functioning as intended with sequential input values.

Min



The labels 'clear', 'in1', 'in2', etc., denote distinct input signals. The corresponding outputs 'cleared', 'out1', 'out2', etc., are reflected after a delay of two clock cycles. For instance, at the start of the simulation, the 'clear' signal is activated (logic high), leading to the initialization of the output 'min0' through 'min15'. This initialization process completes within two clock cycles, as can be seen at the 'cleared' label, where the output assumes the initial value of 16'sb0111111111111111 (binary representation of 2^15-1 in a 16-bit register).

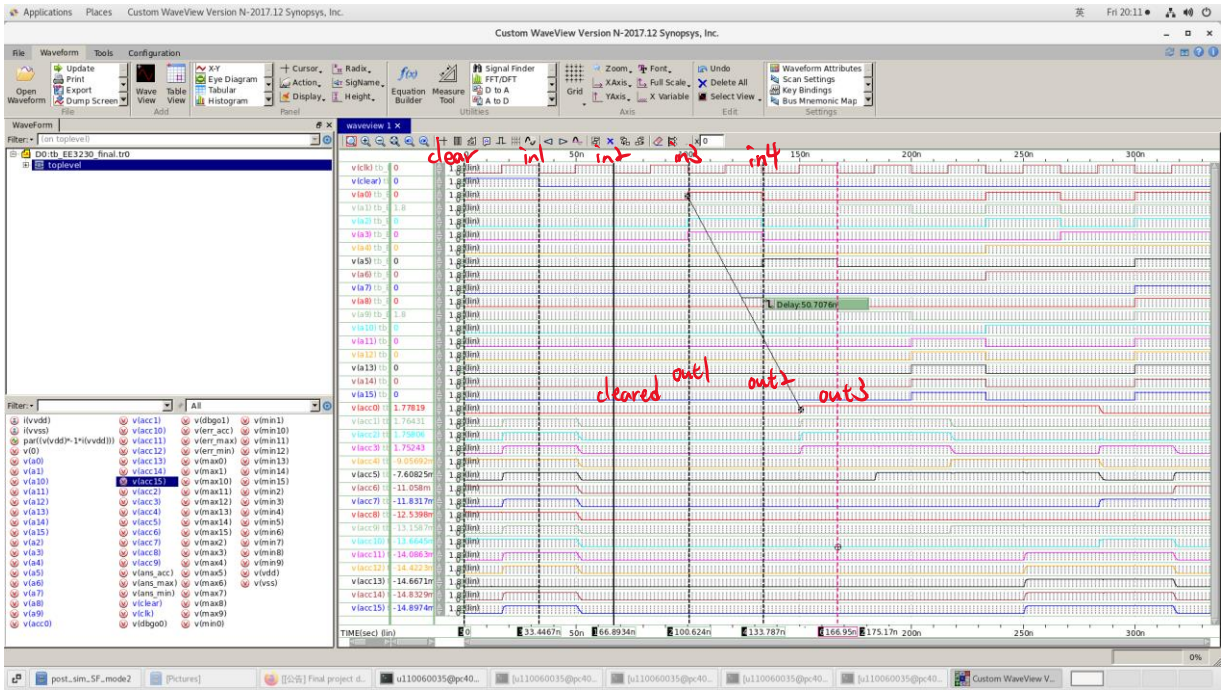
The operating frequency for this simulation is 0.03 GHz (larger than 10MHz = 0.01 GHz), corresponding to a clock period of $\frac{1}{0.03GHz}$, which calculates to approximately 33.333 ns. The observed input-to-output delay is 52.3273 ns, which complies with the requirement that the delay should not exceed twice the clock period (2*tclk). It's important to note that the worst-case delay scenario should approach but not exceed 2*tclk. The larger margin observed here is due to this being a sample case rather than the worst-case scenario.

From the onset, the 'clear' signal initializes the ‘min’ to 0, with 'min' being updated after two clock cycles. The ‘min’ output signal, as observed, correctly accumulates the input values (a0~a7). For the given input sequence of 0, 0, 15, 32, etc., the minimum is always 0, demonstrating that the design is functioning as intended with sequential input values.

5. PostSim (SF)

Mode 2

Acc

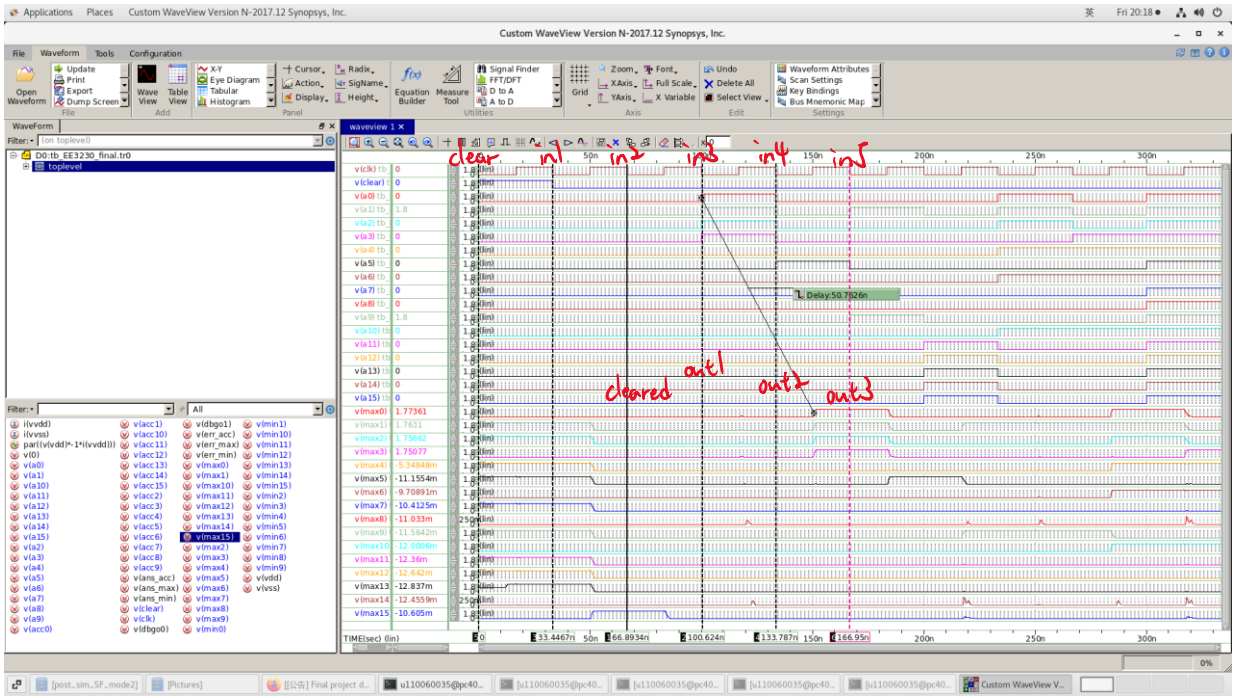


The labels 'clear', 'in1', 'in2', etc., denote distinct input signals. The corresponding outputs 'cleared', 'out1', 'out2', etc., are reflected after a delay of two clock cycles. For instance, at the start of the simulation, the 'clear' signal is activated (logic high), leading to the initialization of the output 'acc0' through 'acc15'. This initialization process completes within two clock cycles, as can be seen at the 'cleared' label, where the output assumes the initial value of 16'sb0 (binary representation of 0 in a 16-bit register).

The operating frequency for this simulation is 0.03 GHz (larger than 10MHz = 0.01 GHz), corresponding to a clock period of $\frac{1}{0.03GHz}$, which calculates to approximately 33.333 ns. The observed input-to-output delay is 50.7076 ns, which complies with the requirement that the delay should not exceed twice the clock period (2*tclk). It's important to note that the worst-case delay scenario should approach but not exceed 2*tclk. The larger margin observed here is due to this being a sample case rather than the worst-case scenario.

From the onset, the 'clear' signal initializes the 'acc' to 0, with 'acc' being updated after two clock cycles. The 'acc' output signal, as observed, correctly accumulates the input values (a0~a7). For the given input sequence of 0, 0, 15, 32 etc., the accumulation sequence for 'acc' is 0, 0, 15, 47 etc., demonstrating that the design is functioning as intended with sequential input values.

Max

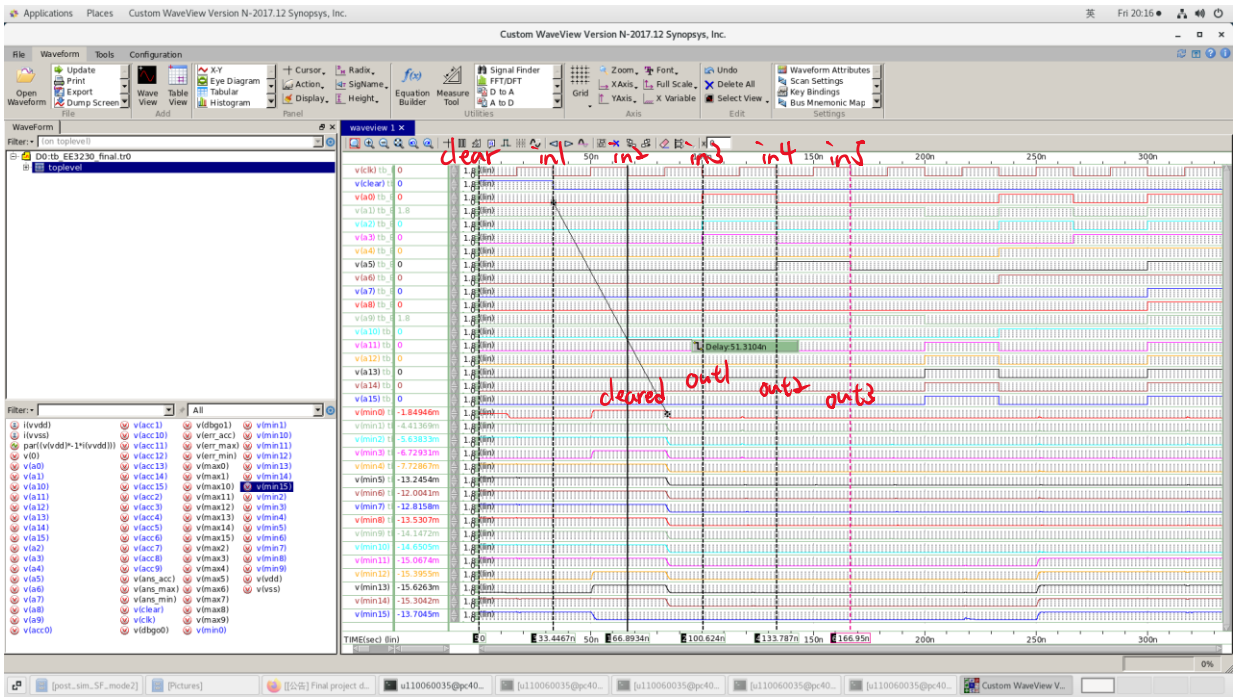


The labels 'clear', 'in1', 'in2', etc., denote distinct input signals. The corresponding outputs 'cleared', 'out1', 'out2', etc., are reflected after a delay of two clock cycles. For instance, at the start of the simulation, the 'clear' signal is activated (logic high), leading to the initialization of the output 'max0' through 'max15'. This initialization process completes within two clock cycles, as can be seen at the 'cleared' label, where the output assumes the initial value of 16'sb1000000000000000 (binary representation of -2^15 in a 16-bit register).

The operating frequency for this simulation is 0.03 GHz (larger than 10MHz = 0.01 GHz), corresponding to a clock period of $\frac{1}{0.03GHz}$, which calculates to approximately 33.333 ns. The observed input-to-output delay is 50.7626 ns, which complies with the requirement that the delay should not exceed twice the clock period (2*tclk). It's important to note that the worst-case delay scenario should approach but not exceed 2*tclk. The larger margin observed here is due to this being a sample case rather than the worst-case scenario.

From the onset, the 'clear' signal initializes the 'max' to 0, with ‘max’ being updated after two clock cycles. The 'max' output signal, as observed, correctly accumulates the input values (a0~a7). For the given input sequence of 0, 0, 15, 32, etc., the maximum sequence for 'max' is 0, 0, 15, 32, etc., demonstrating that the design is functioning as intended with sequential input values.

Min



The labels 'clear', 'in1', 'in2', etc., denote distinct input signals. The corresponding outputs 'cleared', 'out1', 'out2', etc., are reflected after a delay of two clock cycles. For instance, at the start of the simulation, the 'clear' signal is activated (logic high), leading to the initialization of the output 'min0' through 'min15'. This initialization process completes within two clock cycles, as can be seen at the 'cleared' label, where the output assumes the initial value of 16'sb0111111111111111 (binary representation of 2^15-1 in a 16-bit register).

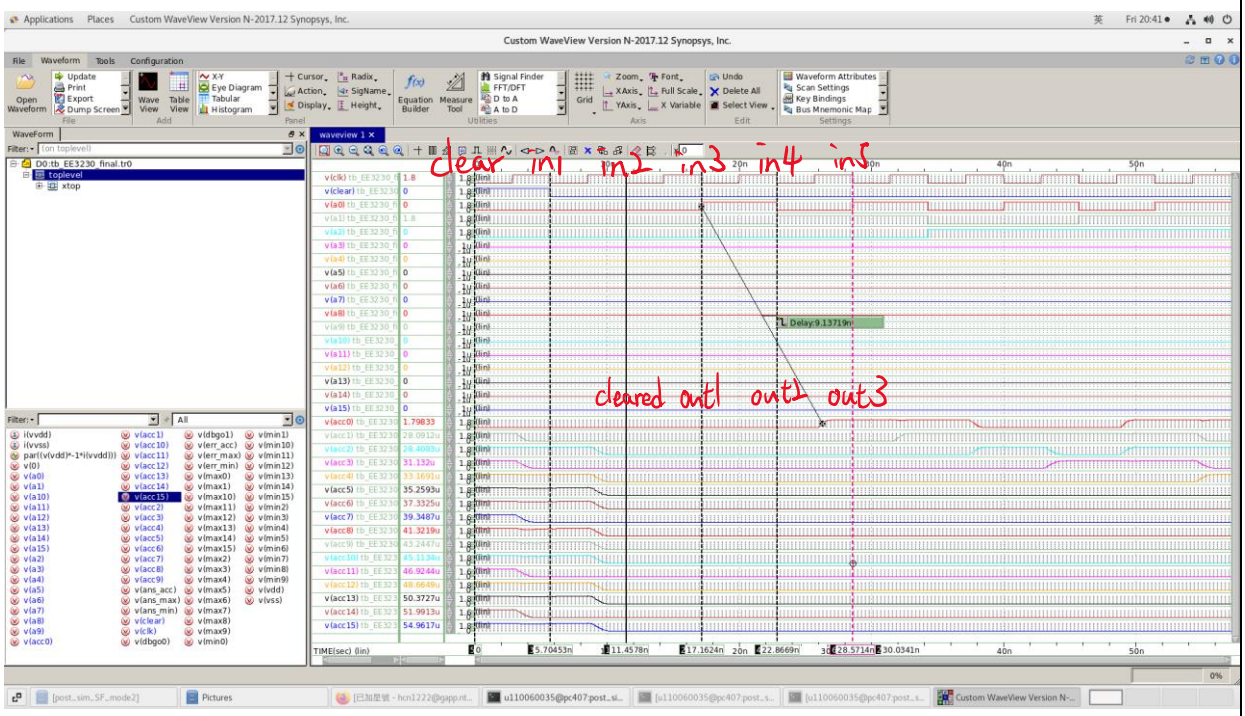
The operating frequency for this simulation is 0.03 GHz (larger than 10MHz = 0.01 GHz), corresponding to a clock period of $\frac{1}{0.03GHz}$, which calculates to approximately 33.333 ns. The observed input-to-output delay is 51.3104 ns, which complies with the requirement that the delay should not exceed twice the clock period (2*tclk). It's important to note that the worst-case delay scenario should approach but not exceed 2*tclk. The larger margin observed here is due to this being a sample case rather than the worst-case scenario.

From the onset, the 'clear' signal initializes the ‘min’ to 0, with 'min' being updated after two clock cycles. The ‘min’ output signal, as observed, correctly accumulates the input values (a0~a7). For the given input sequence of 0, 0, 15, 32, etc., the minimum is always 0, demonstrating that the design is functioning as intended with sequential input values.

6. PostSim (TT)

Mode 0

Acc

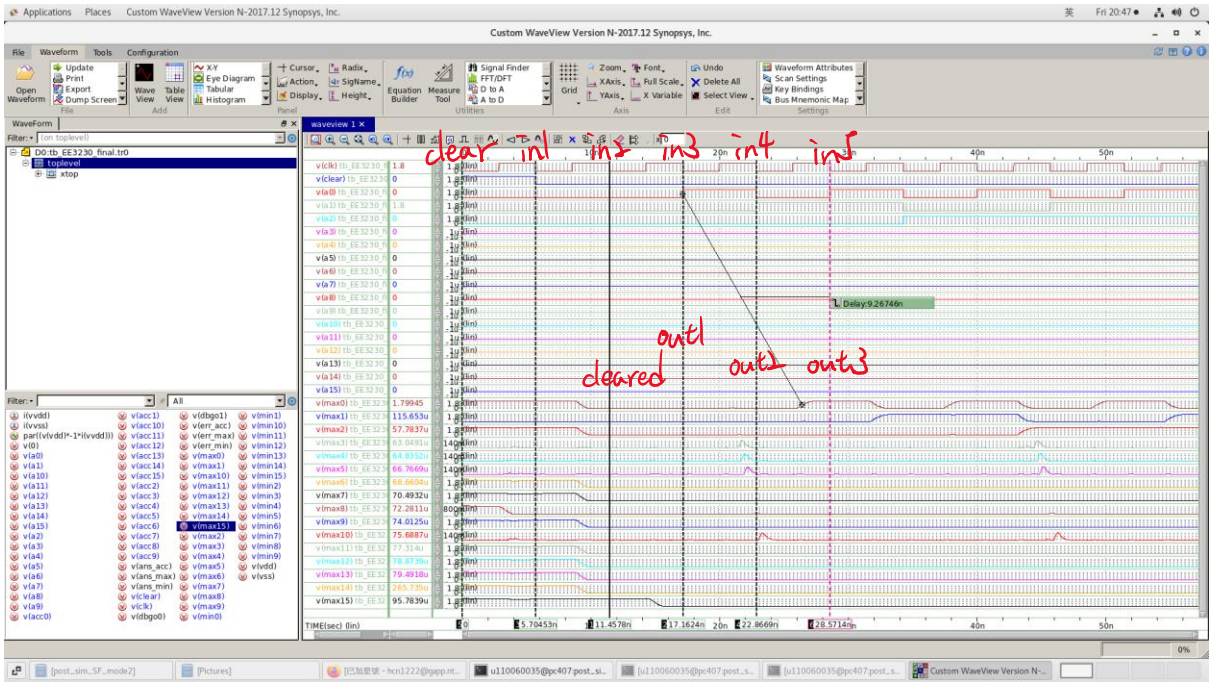


The labels 'clear', 'in1', 'in2', etc., denote distinct input signals. The corresponding outputs 'cleared', 'out1', 'out2', etc., are reflected after a delay of two clock cycles. For instance, at the start of the simulation, the 'clear' signal is activated (logic high), leading to the initialization of the output 'acc0' through 'acc15'. This initialization process completes within two clock cycles, as can be seen at the 'cleared' label, where the output assumes the initial value of 16'sb0 (binary representation of 0 in a 16-bit register).

The maximum operating frequency for our design is 0.175 GHz, corresponding to a clock period of $\frac{1}{0.175GHz}$, which calculates to approximately 5.714 ns. The observed input-to-output delay is 9.13719 ns, which complies with the requirement that the delay should not exceed twice the clock period (2*tclk). It's important to note that the worst-case delay scenario should approach but not exceed 2*tclk. The larger margin observed here is due to this being a sample case rather than the worst-case scenario.

From the onset, the 'clear' signal initializes the 'acc' to 0, with 'acc' being updated after two clock cycles. The 'acc' output signal, as observed, correctly accumulates the input values (a0~a7). For the given input sequence of 0, 0, 1, 2, etc., the accumulation sequence for 'acc' is 0, 0, 1, 3, etc., demonstrating that the design is functioning as intended with sequential input values.

Max

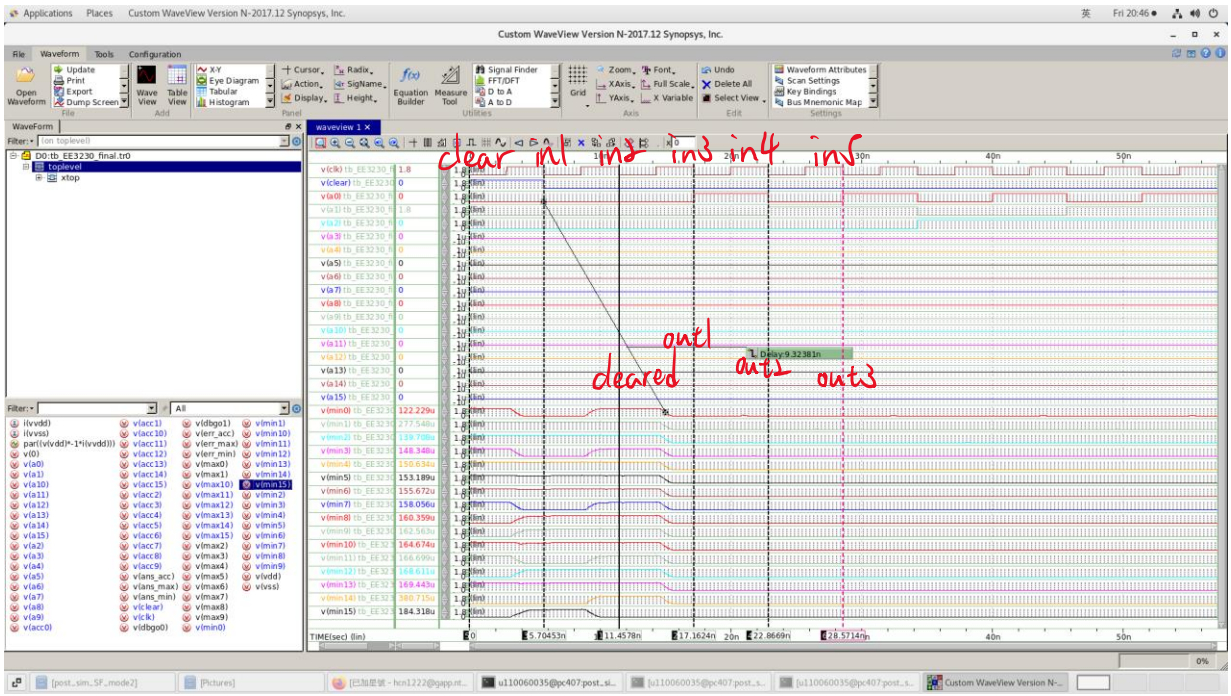


The labels 'clear', 'in1', 'in2', etc., denote distinct input signals. The corresponding outputs 'cleared', 'out1', 'out2', etc., are reflected after a delay of two clock cycles. For instance, at the start of the simulation, the 'clear' signal is activated (logic high), leading to the initialization of the output 'max0' through 'max15'. This initialization process completes within two clock cycles, as can be seen at the 'cleared' label, where the output assumes the initial value of 16'sb1000000000000000 (binary representation of -2^15 in a 16-bit register).

The maximum operating frequency for our design is 0.175 GHz, corresponding to a clock period of $\frac{1}{0.175GHz}$, which calculates to approximately 5.714 ns. The observed input-to-output delay is 9.26746 ns, which complies with the requirement that the delay should not exceed twice the clock period (2*tclk). It's important to note that the worst-case delay scenario should approach but not exceed 2*tclk. The larger margin observed here is due to this being a sample case rather than the worst-case scenario.

From the onset, the 'clear' signal initializes the 'max' to 0, with ‘max’ being updated after two clock cycles. The 'max' output signal, as observed, correctly accumulates the input values (a0~a7). For the given input sequence of 0, 0, 1, 2, etc., the maximum sequence for 'max' is 0, 0, 1, 2, etc., demonstrating that the design is functioning as intended with sequential input values.

Min



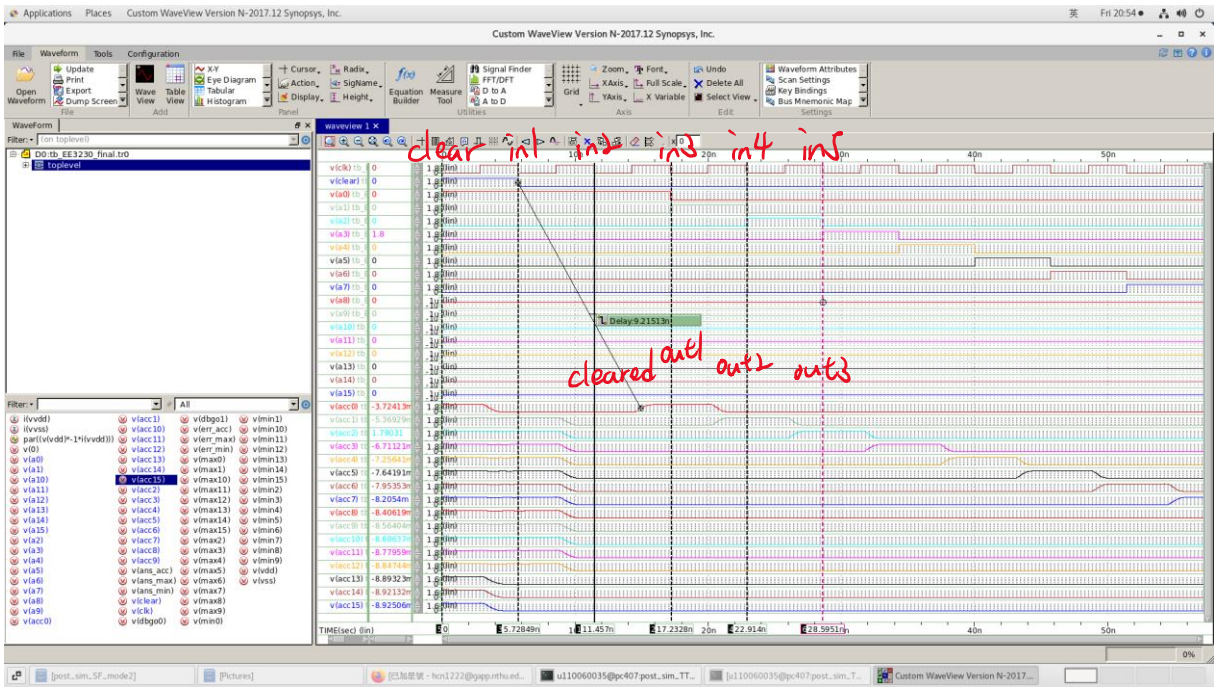
The labels 'clear', 'in1', 'in2', etc., denote distinct input signals. The corresponding outputs 'cleared', 'out1', 'out2', etc., are reflected after a delay of two clock cycles. For instance, at the start of the simulation, the 'clear' signal is activated (logic high), leading to the initialization of the output 'min0' through 'min15'. This initialization process completes within two clock cycles, as can be seen at the 'cleared' label, where the output assumes the initial value of 16'sb0111111111111111 (binary representation of 2^15-1 in a 16-bit register).

The maximum operating frequency for our design is 0.175 GHz, corresponding to a clock period of $\frac{1}{0.175GHz}$, which calculates to approximately 5.714 ns. The observed input-to-output delay is 9.32381 ns, which complies with the requirement that the delay should not exceed twice the clock period (2*tclk). It's important to note that the worst-case delay scenario should approach but not exceed 2*tclk. The larger margin observed here is due to this being a sample case rather than the worst-case scenario.

From the onset, the 'clear' signal initializes the ‘min’ to 0, with 'min' being updated after two clock cycles. The ‘min’ output signal, as observed, correctly accumulates the input values (a0~a7). For the given input sequence of 0, 0, 1, 2, etc., the minimum is always 0, demonstrating that the design is functioning as intended with sequential input values.

Mode 1

Acc



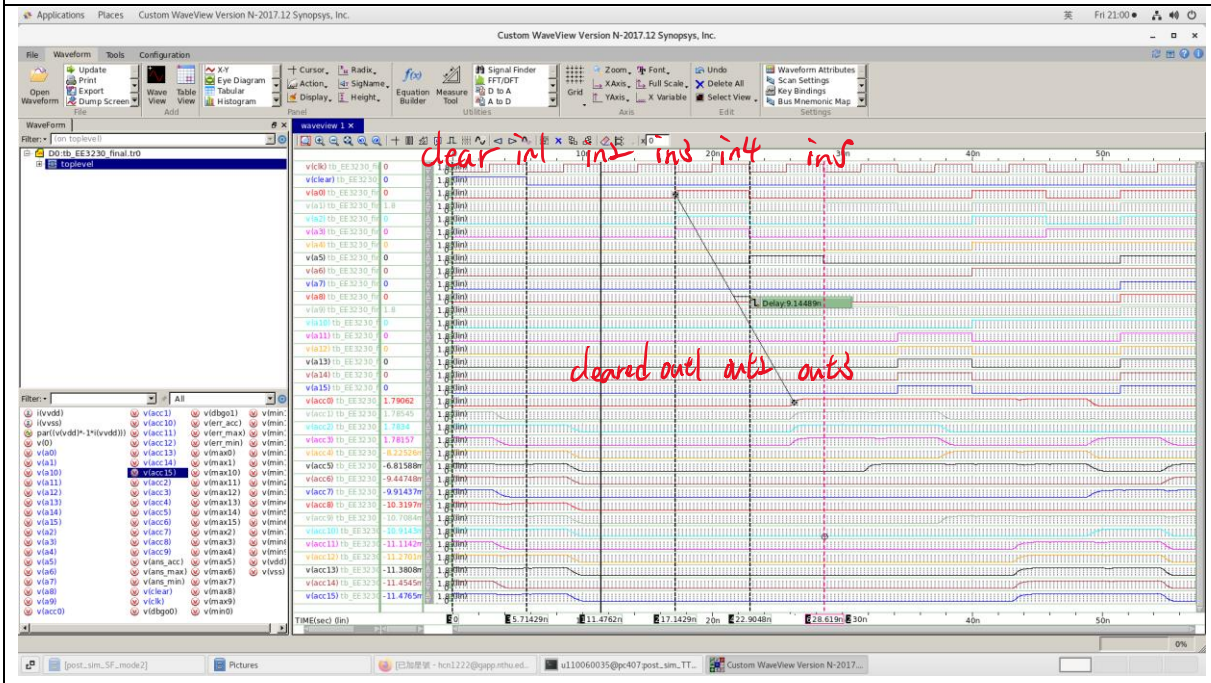
The labels 'clear', 'in1', 'in2', etc., denote distinct input signals. The corresponding outputs 'cleared', 'out1', 'out2', etc., are reflected after a delay of two clock cycles. For instance, at the start of the simulation, the 'clear' signal is activated (logic high), leading to the initialization of the output 'acc0' through 'acc15'. This initialization process completes within two clock cycles, as can be seen at the 'cleared' label, where the output assumes the initial value of 16'sb0 (binary representation of 0 in a 16-bit register).

The maximum operating frequency for our design is 0.175 GHz, corresponding to a clock period of $\frac{1}{0.175GHz}$, which calculates to approximately 5.714 ns. The observed input-to-output delay is 9.21513 ns, which complies with the requirement that the delay should not exceed twice the clock period (2*tclk). It's important to note that the worst-case delay scenario should approach but not exceed 2*tclk. The larger margin observed here is due to this being a sample case rather than the worst-case scenario.

From the onset, the 'clear' signal initializes the 'acc' to 0, with 'acc' being updated after two clock cycles. The 'acc' output signal, as observed, correctly accumulates the input values (a0~a7). For the given input sequence of 1, 1, 2, 4, 8 etc., the accumulation sequence for 'acc' is 1, 2, 4, 8. 16, etc., demonstrating that the design is functioning as intended with sequential input values.

Mode 2

Acc

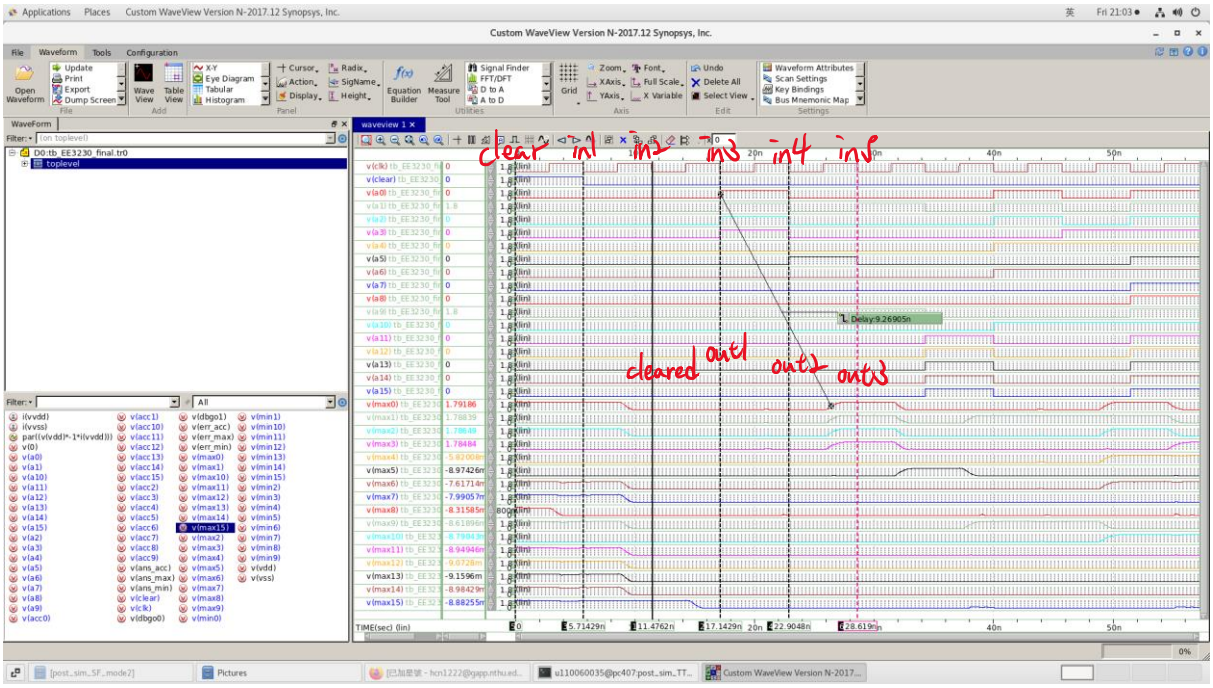


The labels 'clear', 'in1', 'in2', etc., denote distinct input signals. The corresponding outputs 'cleared', 'out1', 'out2', etc., are reflected after a delay of two clock cycles. For instance, at the start of the simulation, the 'clear' signal is activated (logic high), leading to the initialization of the output 'acc0' through 'acc15'. This initialization process completes within two clock cycles, as can be seen at the 'cleared' label, where the output assumes the initial value of 16'sb0 (binary representation of 0 in a 16-bit register).

The maximum operating frequency for our design is 0.175 GHz, corresponding to a clock period of $\frac{1}{0.175GHz}$, which calculates to approximately 5.714 ns. The observed input-to-output delay is 9.14489 ns, which complies with the requirement that the delay should not exceed twice the clock period (2*tclk). It's important to note that the worst-case delay scenario should approach but not exceed 2*tclk. The larger margin observed here is due to this being a sample case rather than the worst-case scenario.

From the onset, the 'clear' signal initializes the 'acc' to 0, with 'acc' being updated after two clock cycles. The 'acc' output signal, as observed, correctly accumulates the input values (a0~a7). For the given input sequence of 0, 0, 15, 32 etc., the accumulation sequence for 'acc' is 0, 0, 15, 47 etc., demonstrating that the design is functioning as intended with sequential input values.

Max

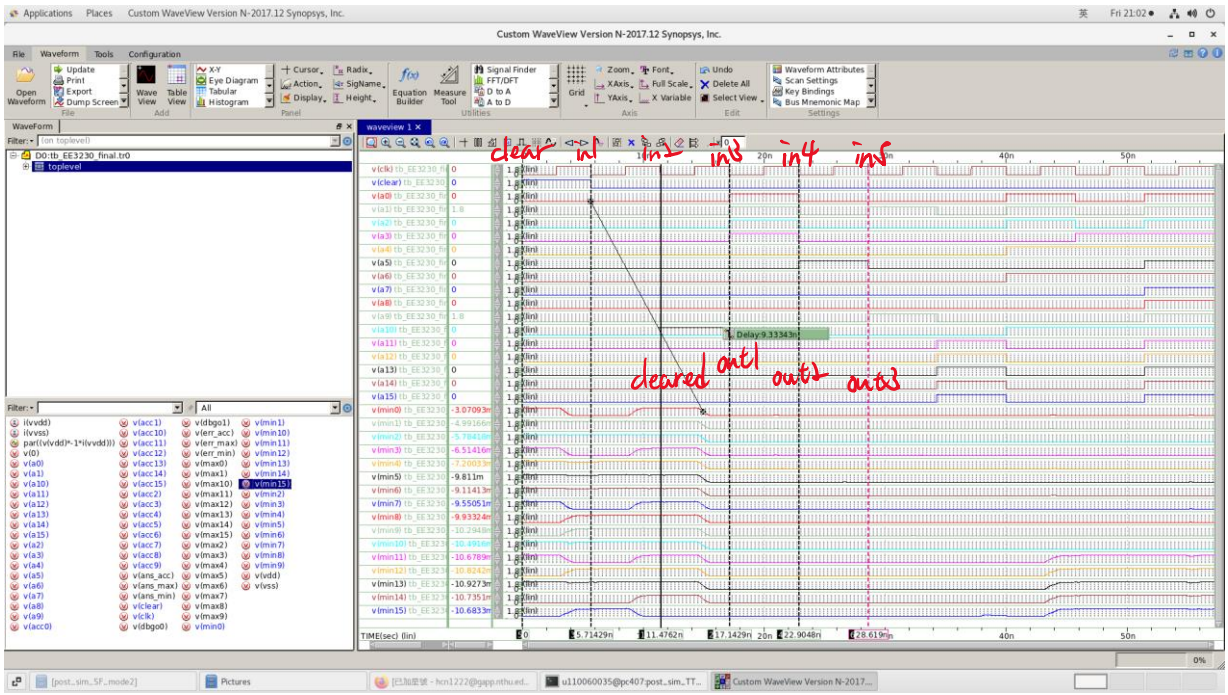


The labels 'clear', 'in1', 'in2', etc., denote distinct input signals. The corresponding outputs 'cleared', 'out1', 'out2', etc., are reflected after a delay of two clock cycles. For instance, at the start of the simulation, the 'clear' signal is activated (logic high), leading to the initialization of the output 'max0' through 'max15'. This initialization process completes within two clock cycles, as can be seen at the 'cleared' label, where the output assumes the initial value of 16'sb1000000000000000 (binary representation of -2^15 in a 16-bit register).

The maximum operating frequency for our design is 0.175 GHz, corresponding to a clock period of $\frac{1}{0.175GHz}$, which calculates to approximately 5.714 ns. The observed input-to-output delay is 9.26905 ns, which complies with the requirement that the delay should not exceed twice the clock period (2*tclk). It's important to note that the worst-case delay scenario should approach but not exceed 2*tclk. The larger margin observed here is due to this being a sample case rather than the worst-case scenario.

From the onset, the 'clear' signal initializes the 'max' to 0, with ‘max’ being updated after two clock cycles. The 'max' output signal, as observed, correctly accumulates the input values (a0~a7). For the given input sequence of 0, 0, 15, 32, etc., the maximum sequence for 'max' is 0, 0, 15, 32, etc., demonstrating that the design is functioning as intended with sequential input values.

Min



The labels 'clear', 'in1', 'in2', etc., denote distinct input signals. The corresponding outputs 'cleared', 'out1', 'out2', etc., are reflected after a delay of two clock cycles. For instance, at the start of the simulation, the 'clear' signal is activated (logic high), leading to the initialization of the output 'min0' through 'min15'. This initialization process completes within two clock cycles, as can be seen at the 'cleared' label, where the output assumes the initial value of 16'sb0111111111111111 (binary representation of 2^15-1 in a 16-bit register).

The maximum operating frequency for our design is 0.175 GHz, corresponding to a clock period of $\frac{1}{0.175GHz}$, which calculates to approximately 5.714 ns. The observed input-to-output delay is 9.33343 ns, which complies with the requirement that the delay should not exceed twice the clock period (2*tclk). It's important to note that the worst-case delay scenario should approach but not exceed 2*tclk. The larger margin observed here is due to this being a sample case rather than the worst-case scenario.

From the onset, the 'clear' signal initializes the ‘min’ to 0, with 'min' being updated after two clock cycles. The ‘min’ output signal, as observed, correctly accumulates the input values (a0~a7). For the given input sequence of 0, 0, 15, 32, etc., the minimum is always 0, demonstrating that the design is functioning as intended with sequential input values.

7. Table (competition)

	Corner	Mode	Area um ²	Freq (MHz)	Power(uW)		FOM($\frac{MHz}{uW*um^2}$)	
					10 * tclk	100 * tclk	10 * tclk	100 * tclk
Pre-simulation	TT 25	0		270	2922.4	3253.3		
		1			2608.7	3078.9		
		2			5234.7	2142.1		
Post-simulation	TT 25	0	37950.2017	175	2616.7	2867.3	$1.76226017*10^{-6}$	$1.60823987*10^{-6}$
		1			2500.8	2755.1	$1.84393242*10^{-6}$	$1.67373460*10^{-6}$
		2			4467.4	1915.8	$1.03221252*10^{-6}$	$2.40698726*10^{-6}$
	FF -40	2		30	766.4873	332.9127		
	FS 25	2		30	789.8736	344.9022		
	SF 25	2		30	822.7986	352.8158		
	SS 125	2		30	836.4974	366.0872		

Discussion

The central philosophy of our design is to minimize the area while simplifying the circuit by limiting the variety of components used, such as NAND and NOR gates, thereby reducing complexity and enhancing uniformity. Our circuit exclusively utilizes three types of components: multiplexers (MUX), inverters, and latches (with flip-flops composed of latches and inverters, the 16-bit adder constructed from muxes and inverters, and the initial block built with muxes). Our multiplexer design employs transmission gates, essentially inverter-latch pairs, meaning that our circuit fundamentally relies on just inverters and latches, significantly simplifying the complexity.

Furthermore, our latches are designed without direct connections to VDD and VSS. This choice makes them more susceptible to noise; however, it effectively reduces static leakage current, thereby lowering power consumption—a deliberate decision made after careful consideration. Furthermore, to ensure signal integrity and circuit stability, within each flip-flop, the D and CLK inputs pass through two inverters before further processing.

Next, we intend to compare the power results between presim and postsim. If we examine the values at each respective max frequency, we surprisingly find that the power during presim is higher than that during postsim. However, a more accurate comparison should consider the energy consumed per calculation, which is the power divided by the max frequency. From the data table, we observe that the power/max frequency for presim in TT25 mode0 is 1.08 x 10^-11 Joules, while for postsim in the same mode, it is 1.50 x 10^-11 Joules. This means that each operation in postsim consumes approximately 50% more energy than in presim. The main reason for this discrepancy is

the parasitic capacitance and resistance introduced during the layout phase. One potential improvement could be to route VDD and VSS using higher metal layers to reduce parasitic resistance.

Regarding the comparison of max frequency between presim and postsim, presim shows a max frequency of 270 MHz while postsim shows 175 MHz. The max frequency allowable in postsim is about 25% less than in presim. This is understandable since there is an inverse relationship between max frequency and delay. Due to the parasitic capacitance and resistance from the layout in postsim, the delay increases, which in turn decreases the max frequency.

Summarizing our discussion, postsim results show about a 25% reduction in max frequency and a 50% increase in power consumption compared to presim, which are acceptable outcomes. Future efforts could focus on employing datapath optimization or improved routing layouts to reduce the discrepancies between presim and postsim results.