**CS303 – Lab#3 Report**

**Name-Surname/SID: Alpaslan Hamzaoğlu/25075**

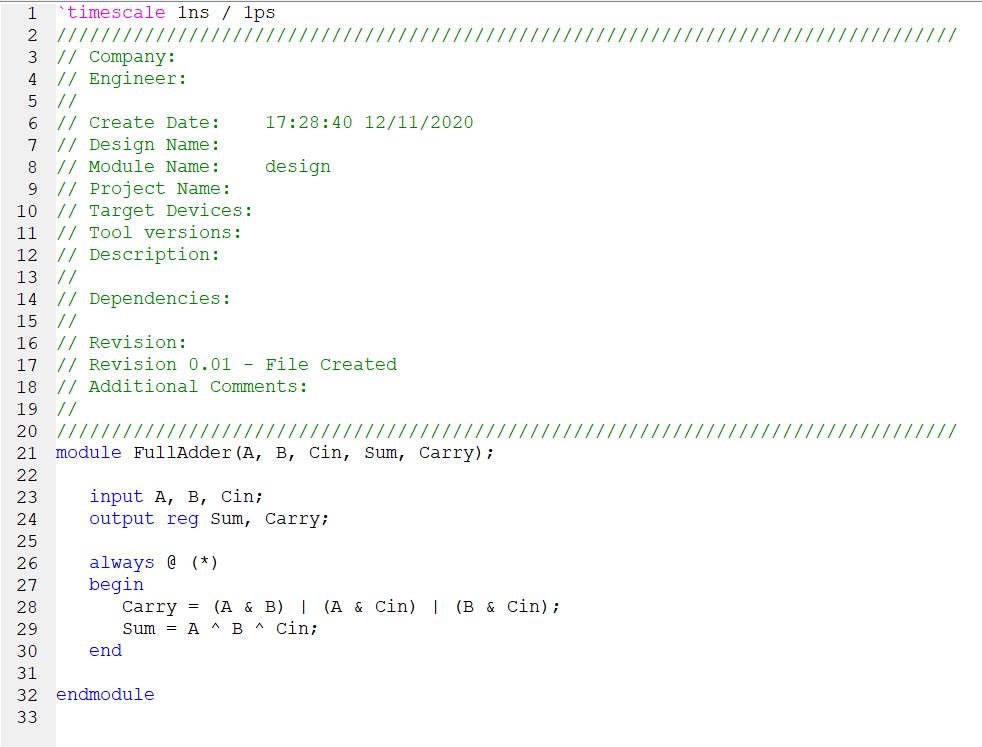
1. **Project Description**

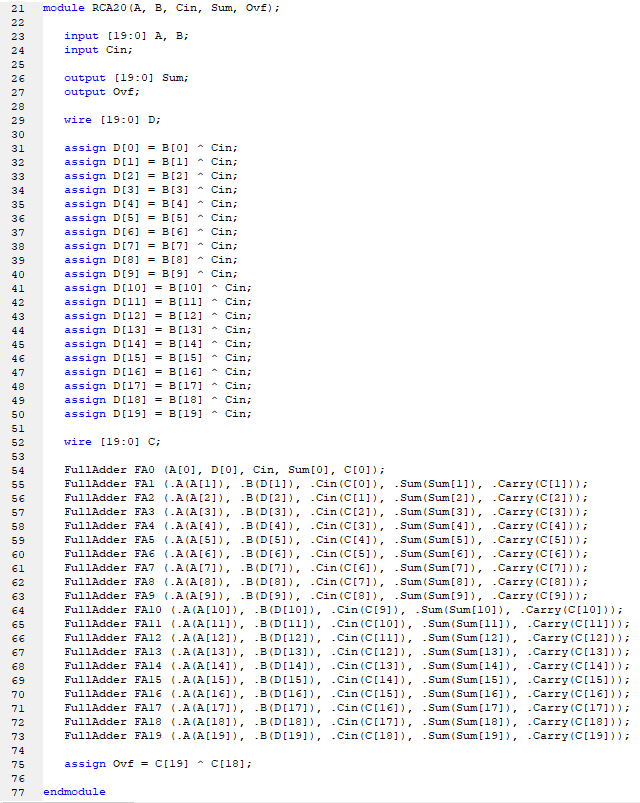
The lab has two main tasks. First one is to implement a 20-bit ripple-carry adder-subtractor using full adders and the second is to implement a 20-bit hybrid adder-subtractor using four 5-bit carry lookahead adders (CLAs) by Verilog. After that there is a simulation part which try to find design is correct. Finally, last part is comparing the two design.

1. **First Design**

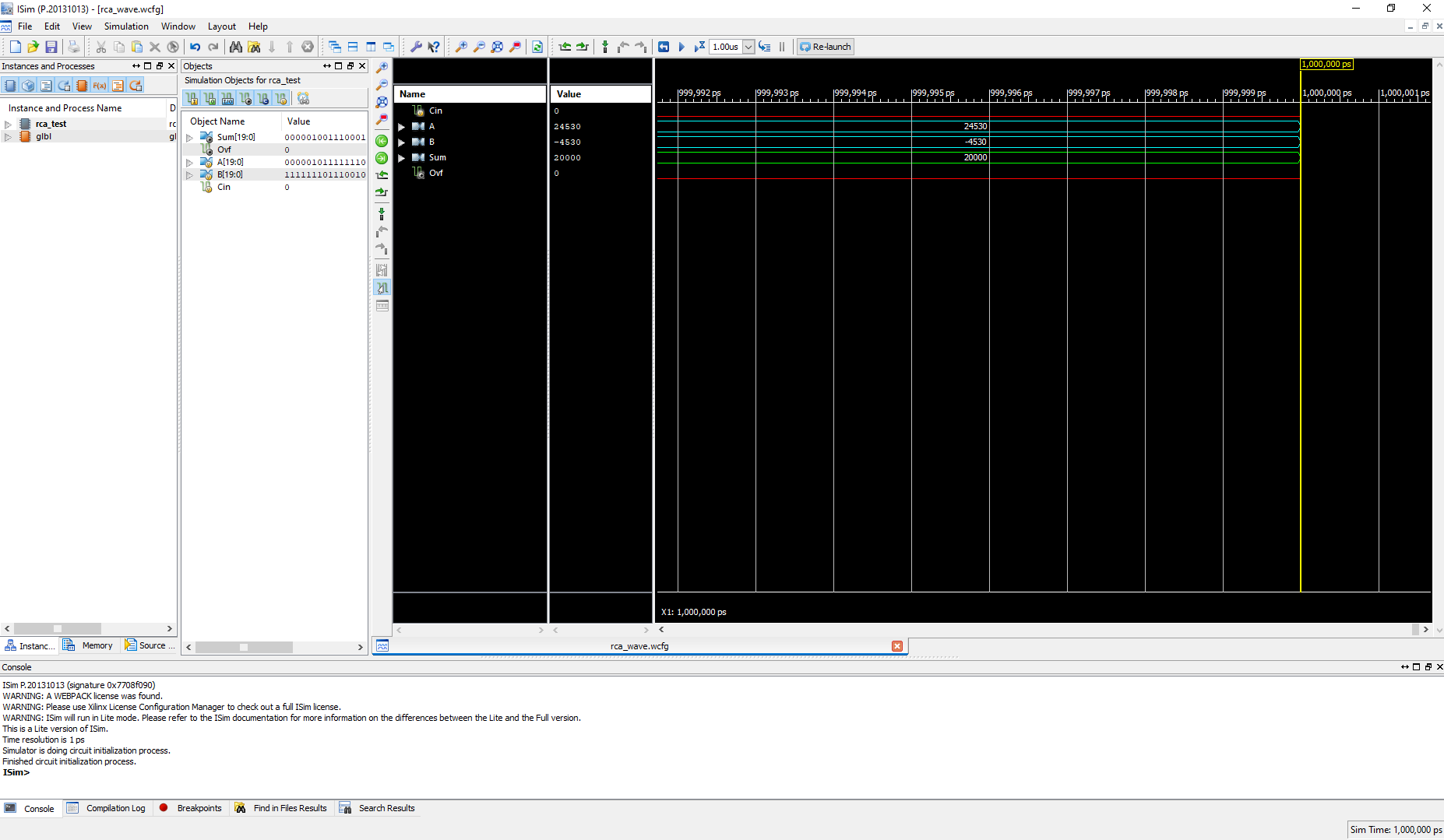
It is the 20-bit ripple-carry adder-subtractor using full adders.

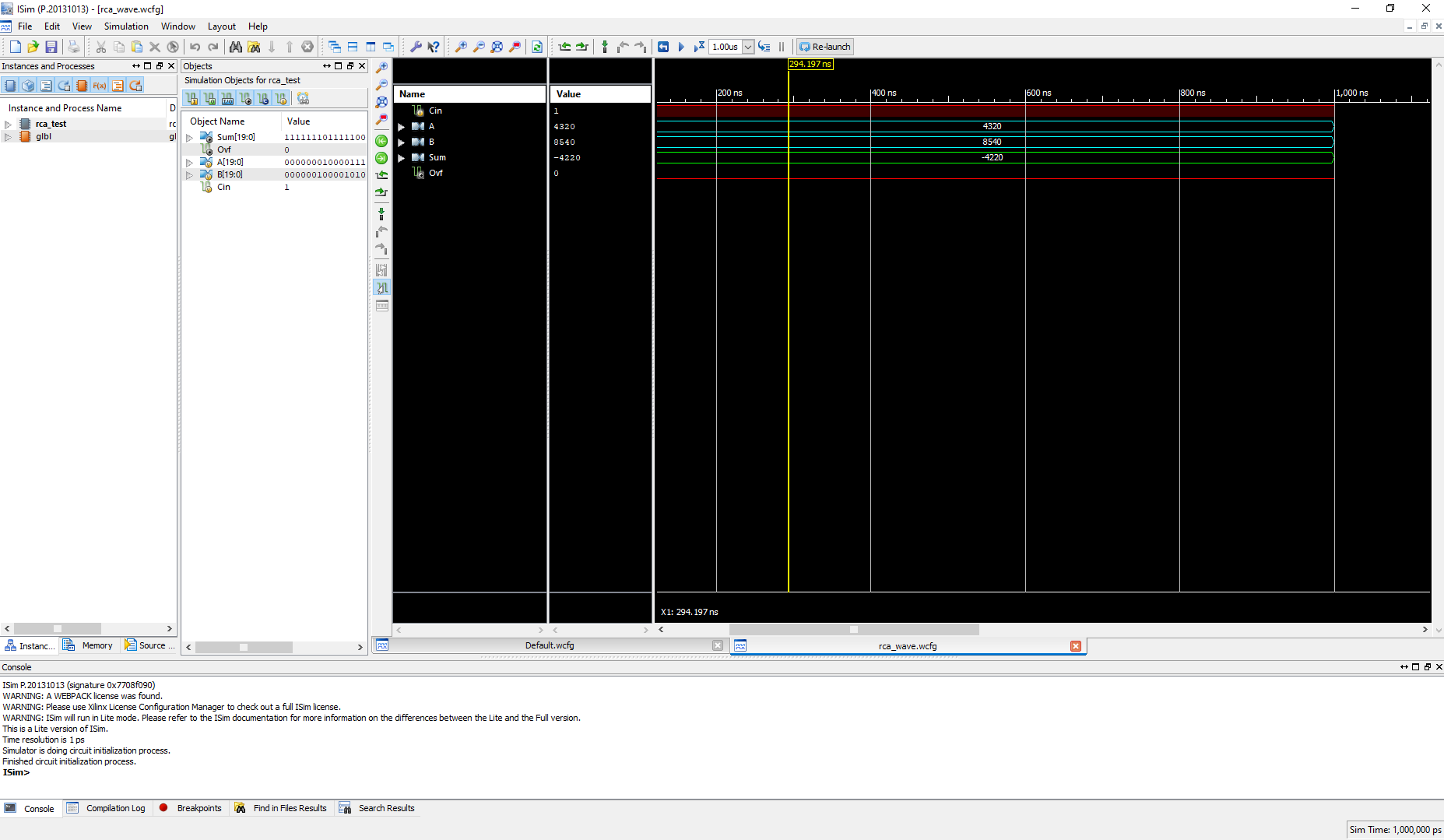
**2.1. Verilog**

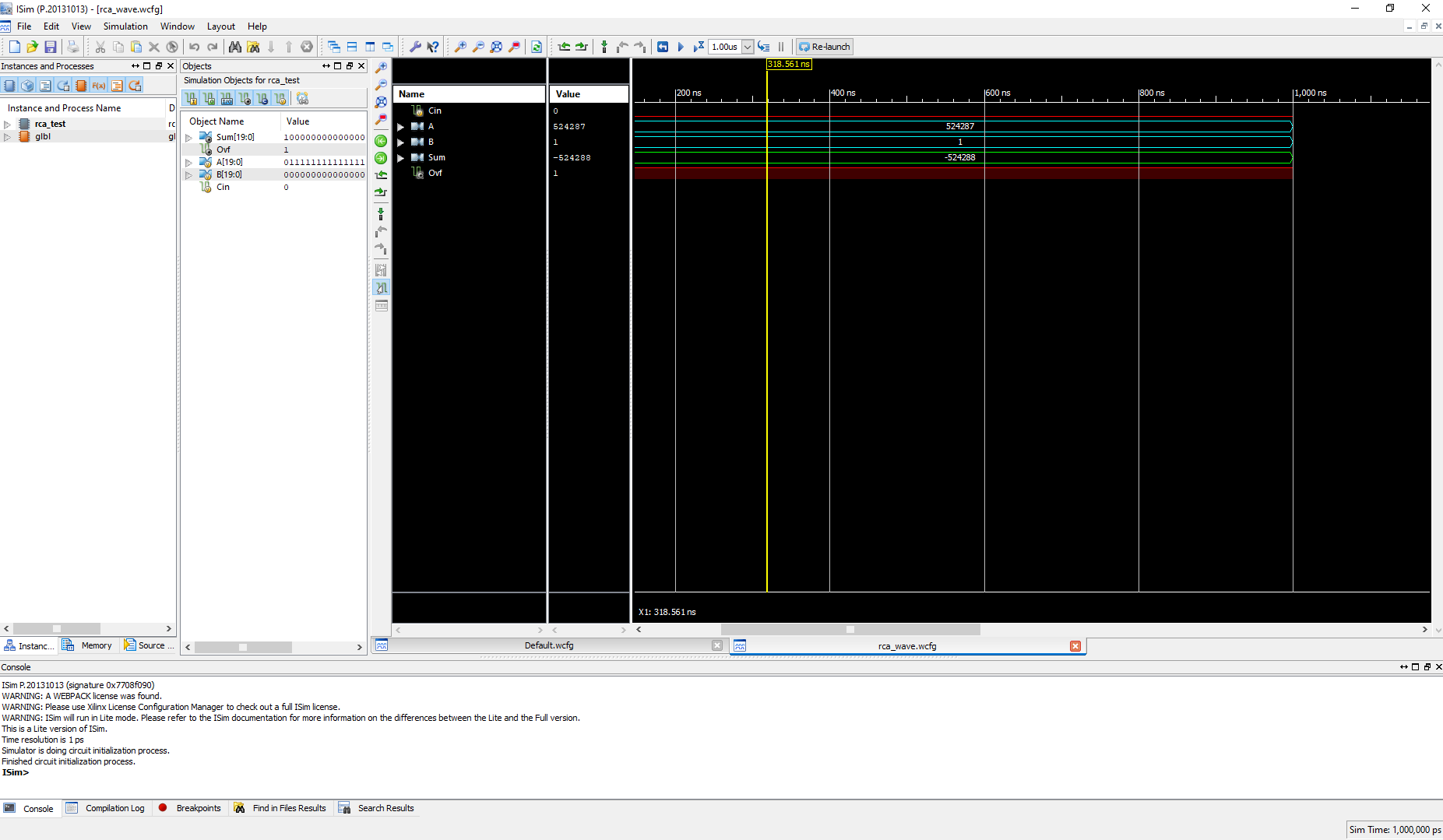
****The first image shows the full adder which is turn into a symbol by name FullAdder.

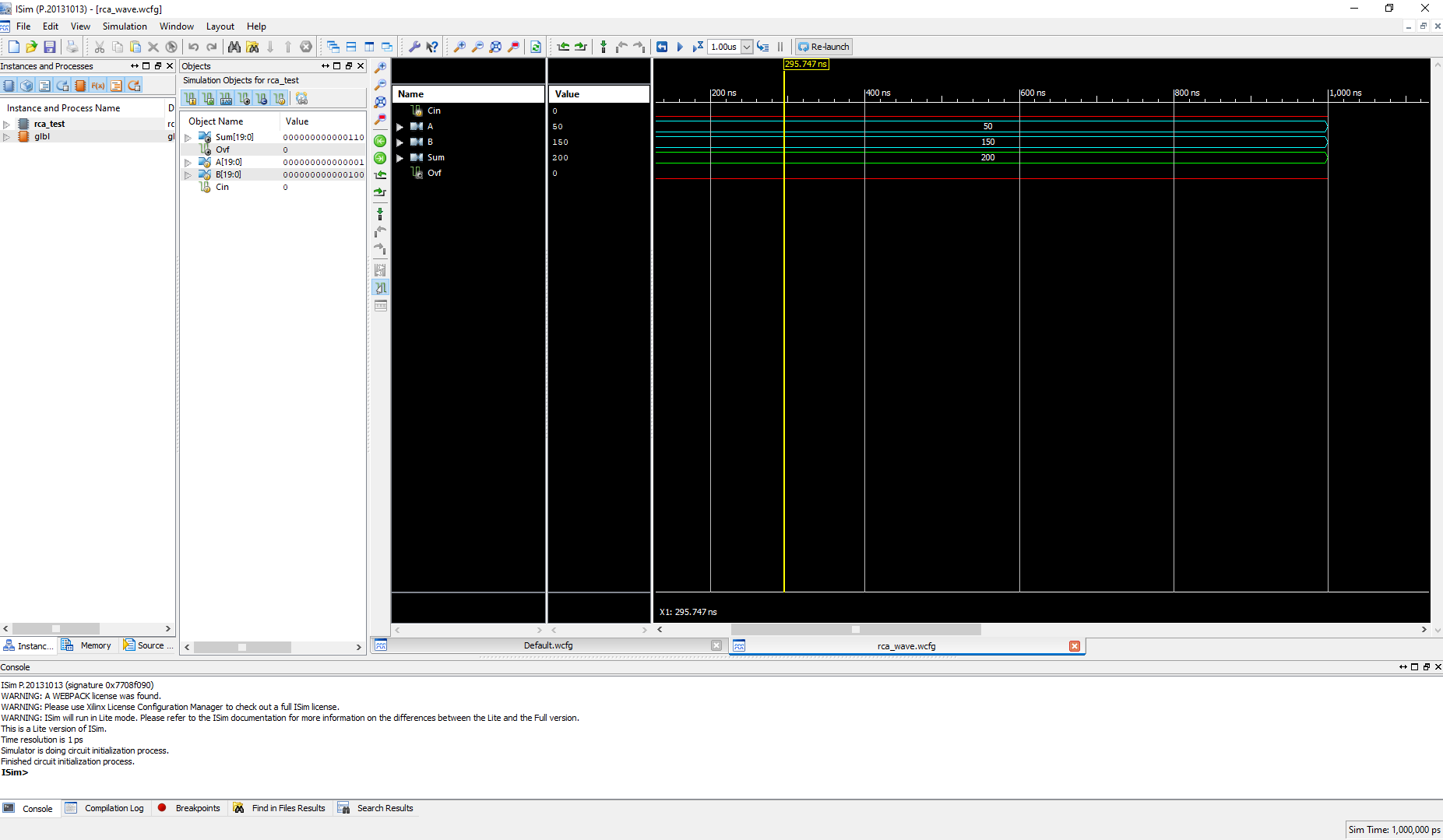
The second image shows the main implementation which contains 20 full adders (FullAdder). There 20 input of A and B, and input of Cin, also 20 output of Sum and one overflow detection (Ovf) output.

**2.2. Simulation**

In the first image, Cin = 0, A = 24530, B = -4530, Sum = 20000 and OVF = 0.

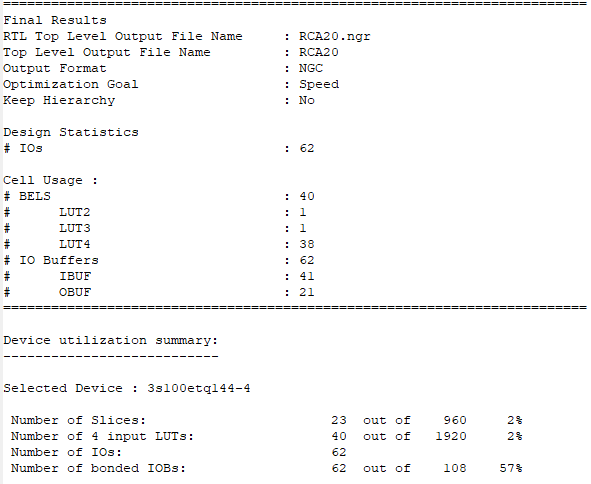
In the second image, Cin = 1, A = 4320, B = 8540, Sum = -4220 and OVF = 0.

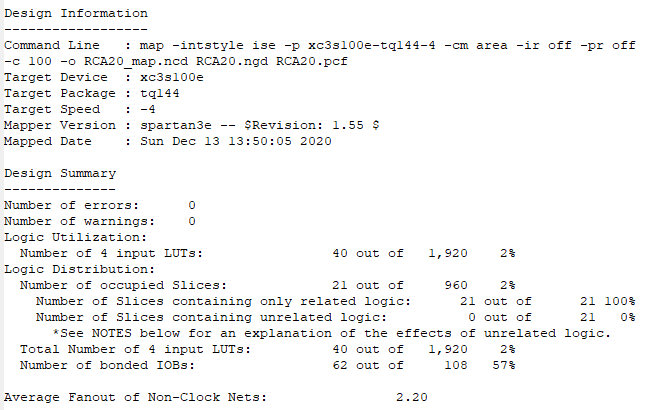
In the third image, Cin = 0, A = 524287, B = 1, Sum = -524288 and OVF = 1.

In the fourth image, Cin = 0, A = 50, B = 150, Sum = 200 and OVF = 0.

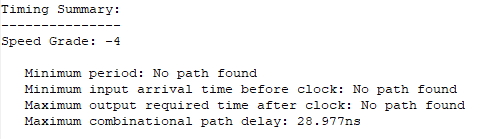
**2.3. Implementation Results**

Synthesis report result for 20-bit ripple-carry adder-subtractor using full adders which includes area and timing estimation.

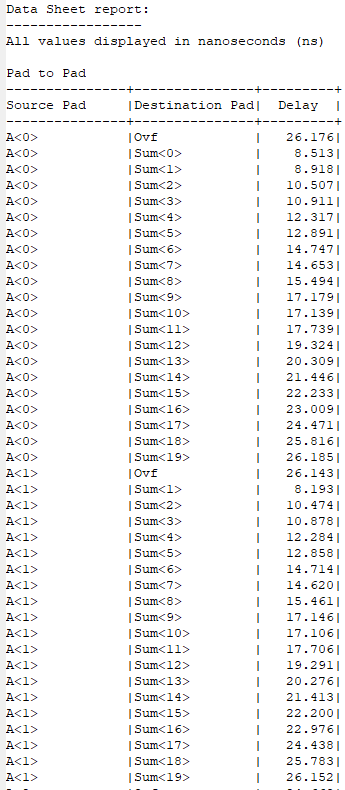
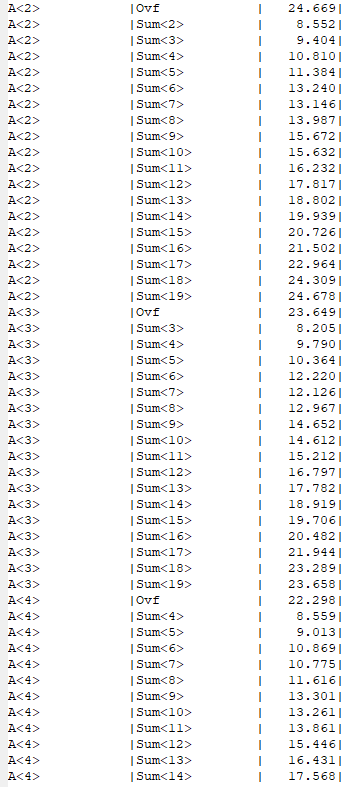
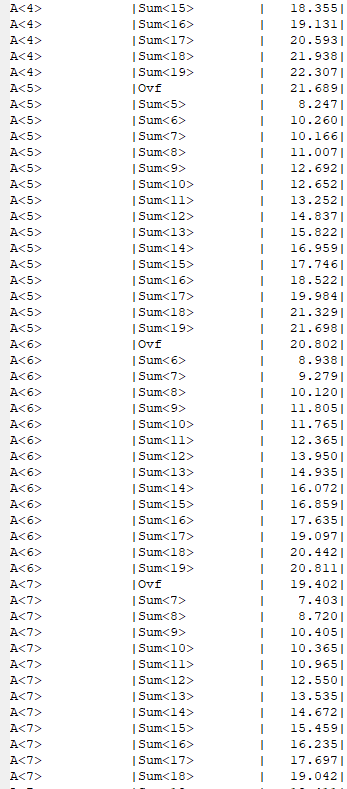
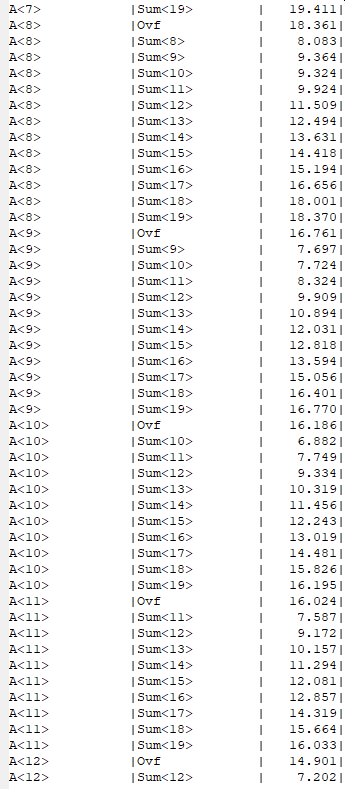


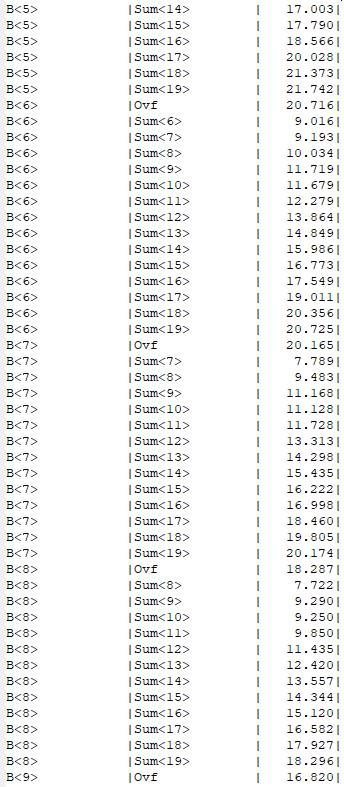
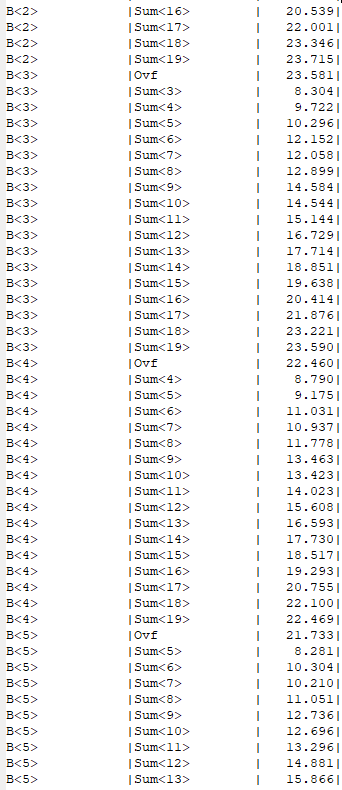
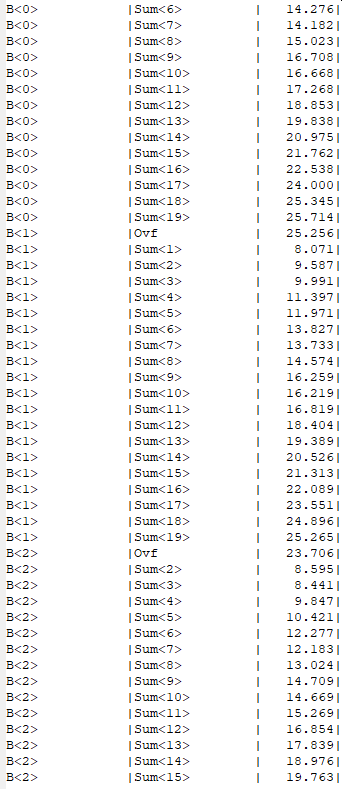
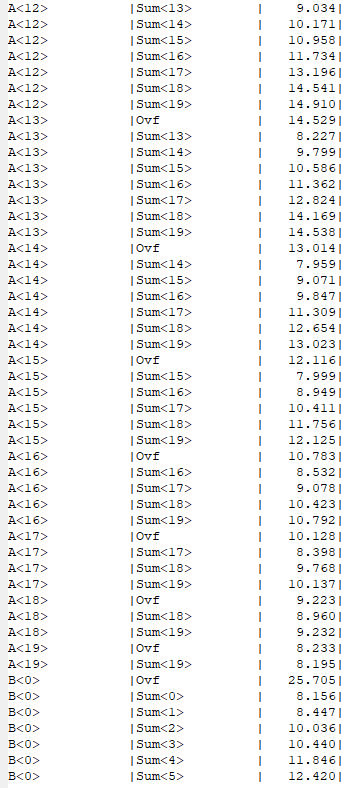
Area for 20-bit ripple-carry adder-subtractor using full adders is 40 (number of 4 input LUTs).

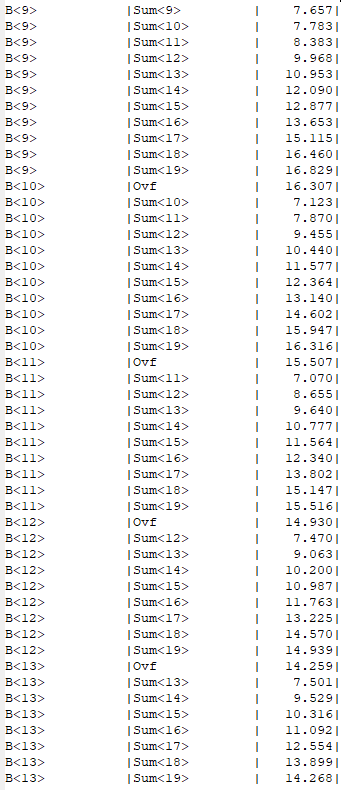
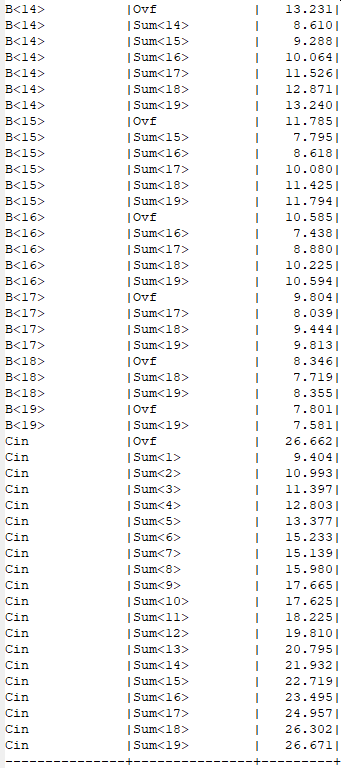
The time for 20-bit ripple-carry adder-subtractor using full adders is 28.977ns (path delay).



Pad to pad for 20-bit ripple-carry adder-subtractor using full adders.



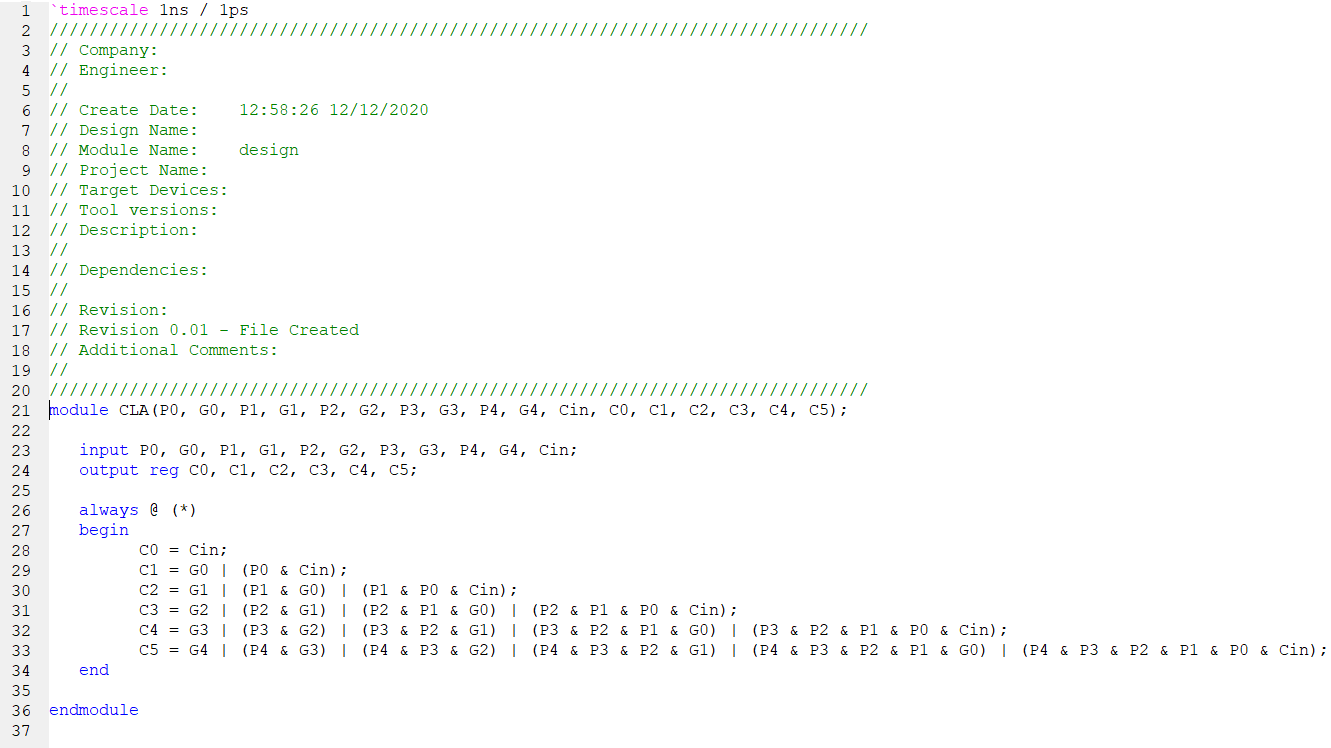


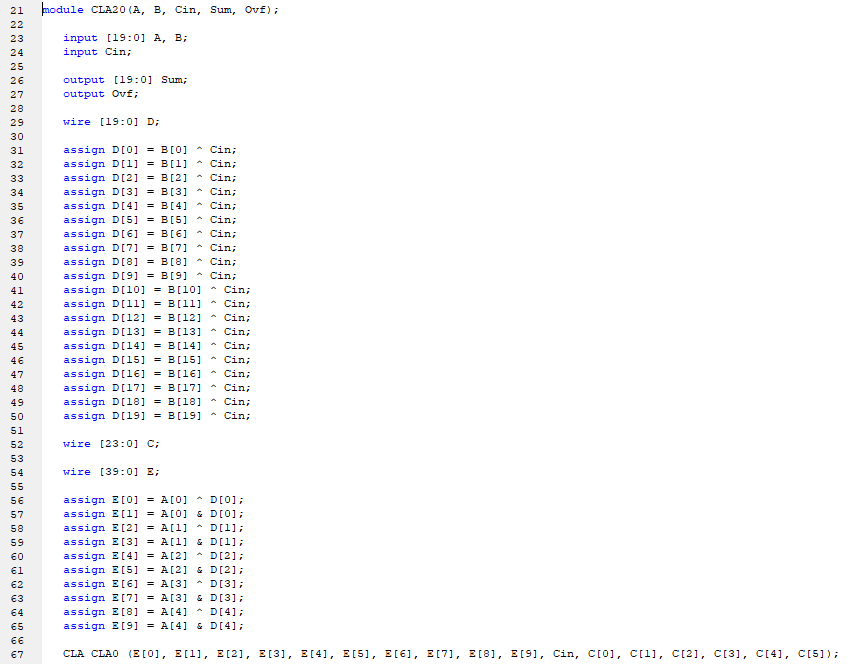


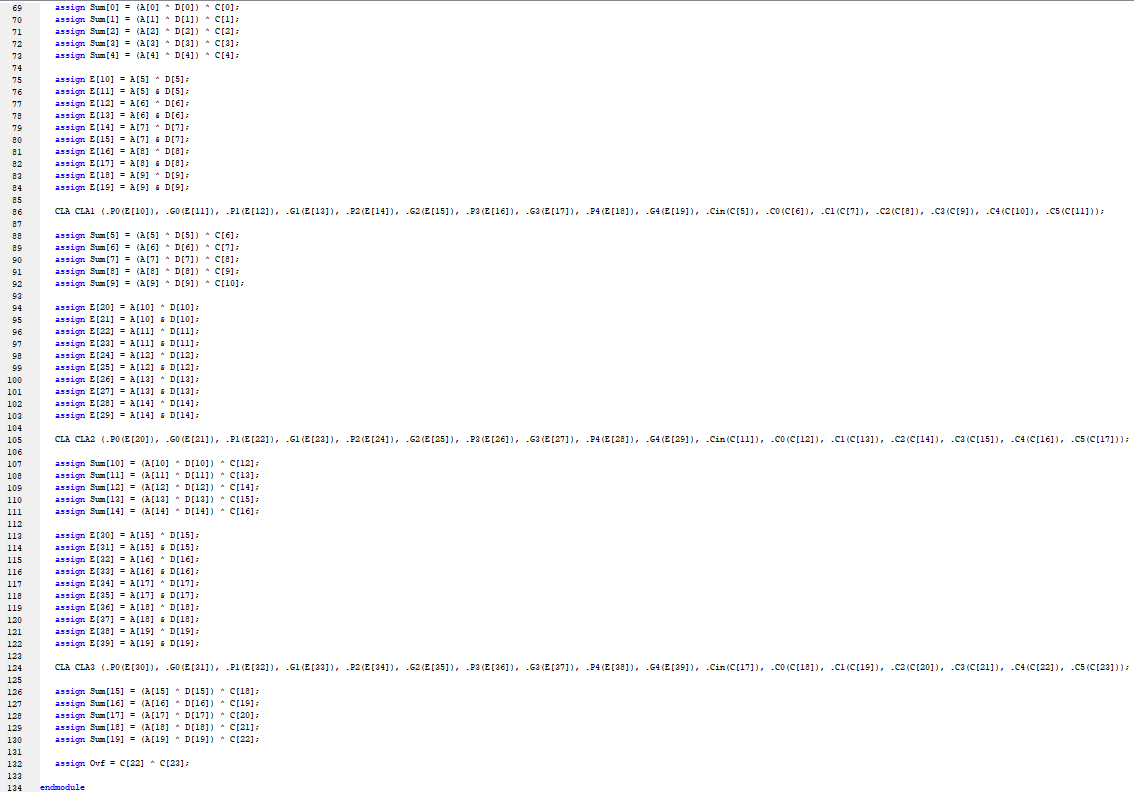
1. **Second Design**

It is the 20-bit hybrid adder-subtractor using four 5-bit carry lookahead adder (CLAs).

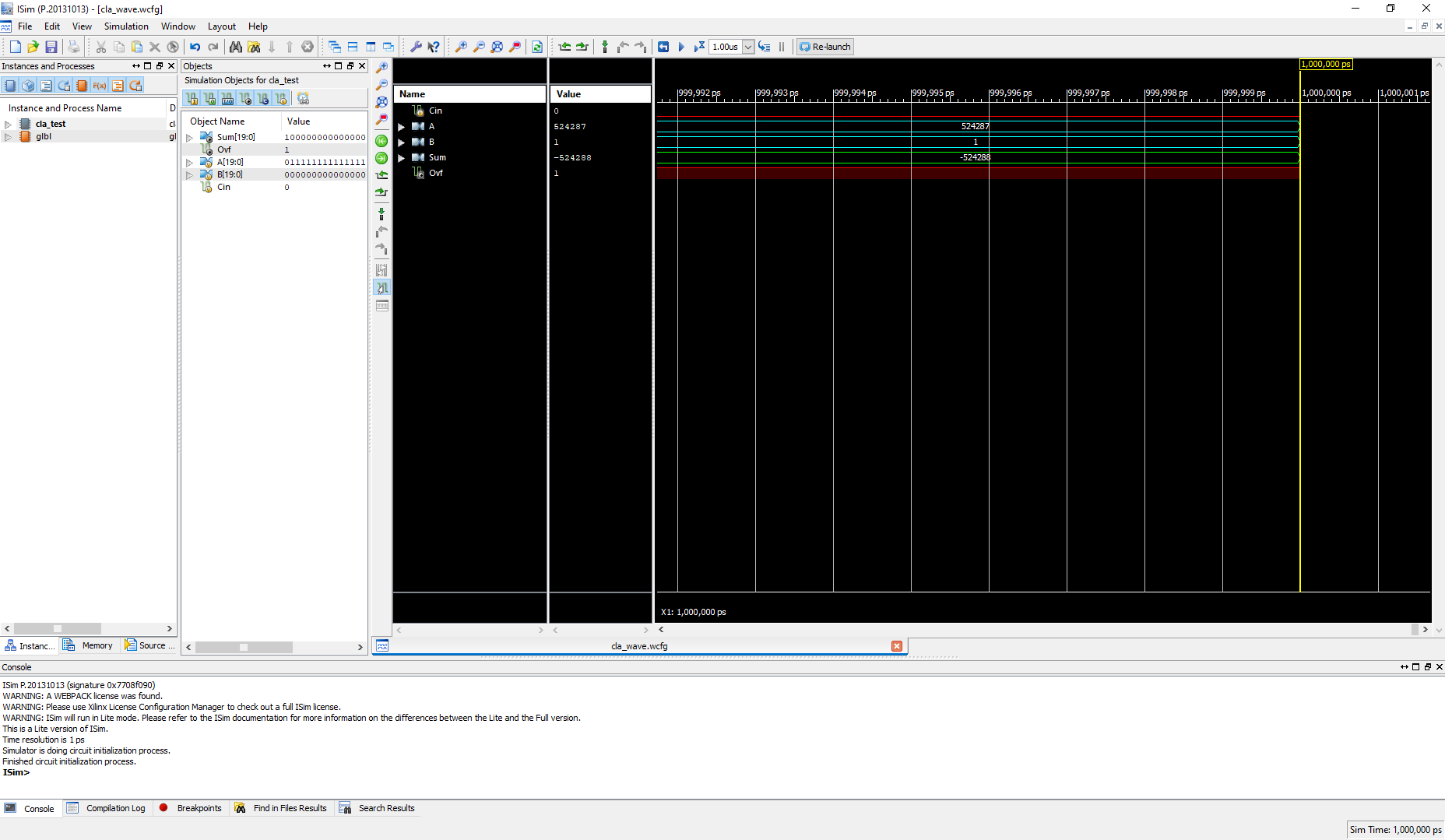
**3.1. Verilog**

The first image shows the 5-bit carry lookahead adder (CLA) which is turn into a symbol by name CLA.

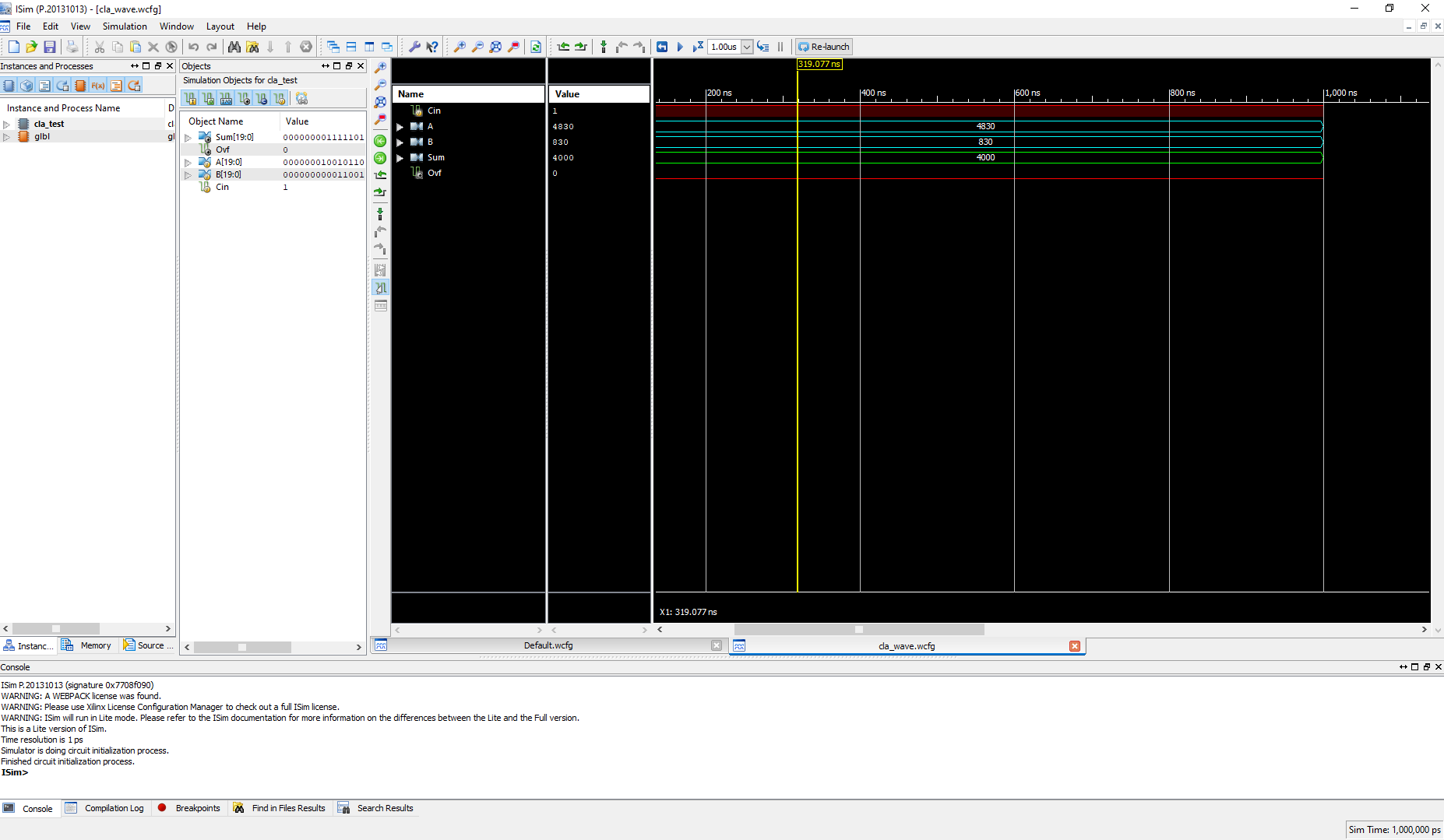
The second and third image shows the main implementation which contains 20 CLA (carry lookahead adder). There 20 input of A and B, and input of Cin, also 20 output of Sum and one overflow detection (Ovf) output.

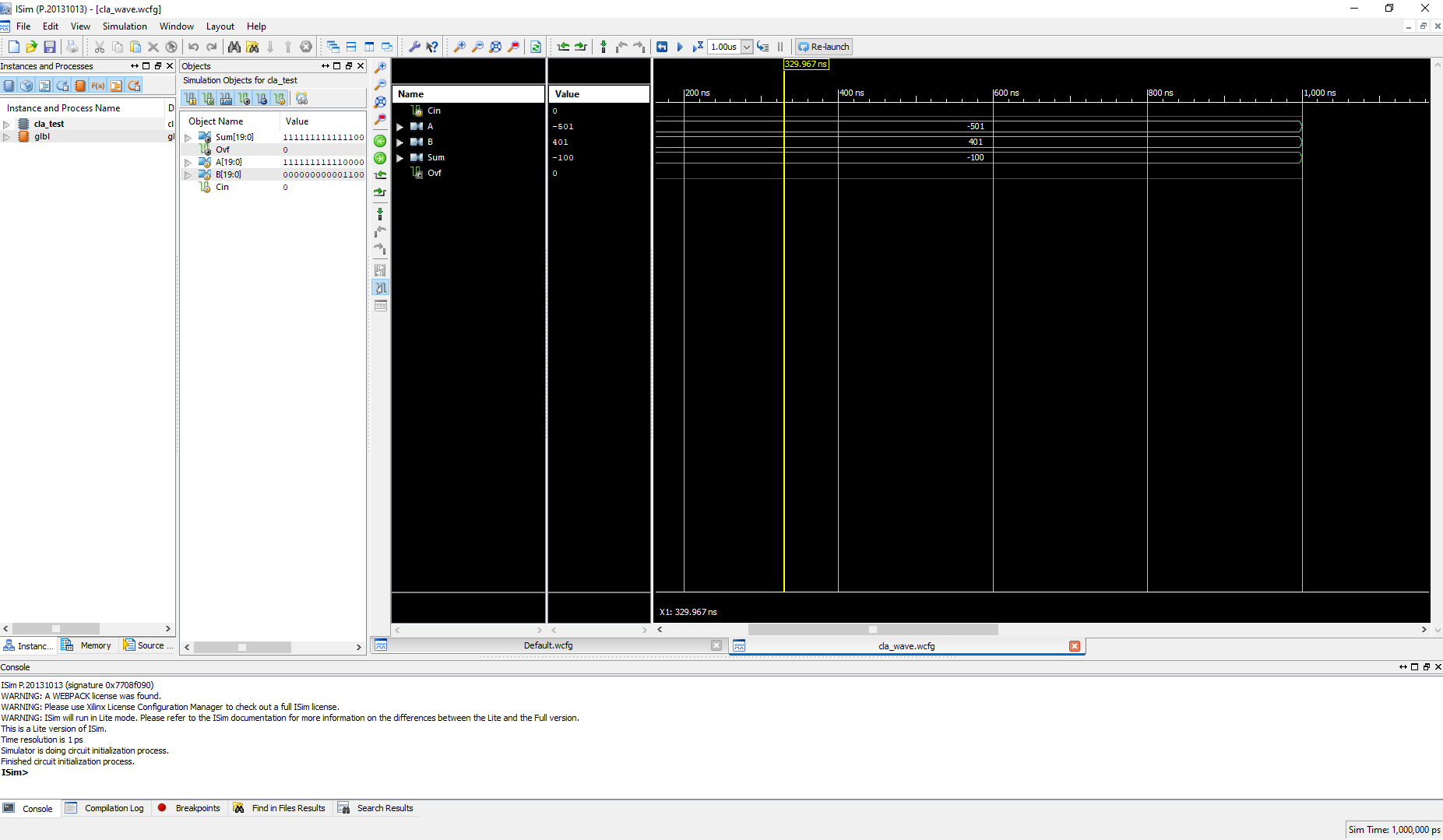


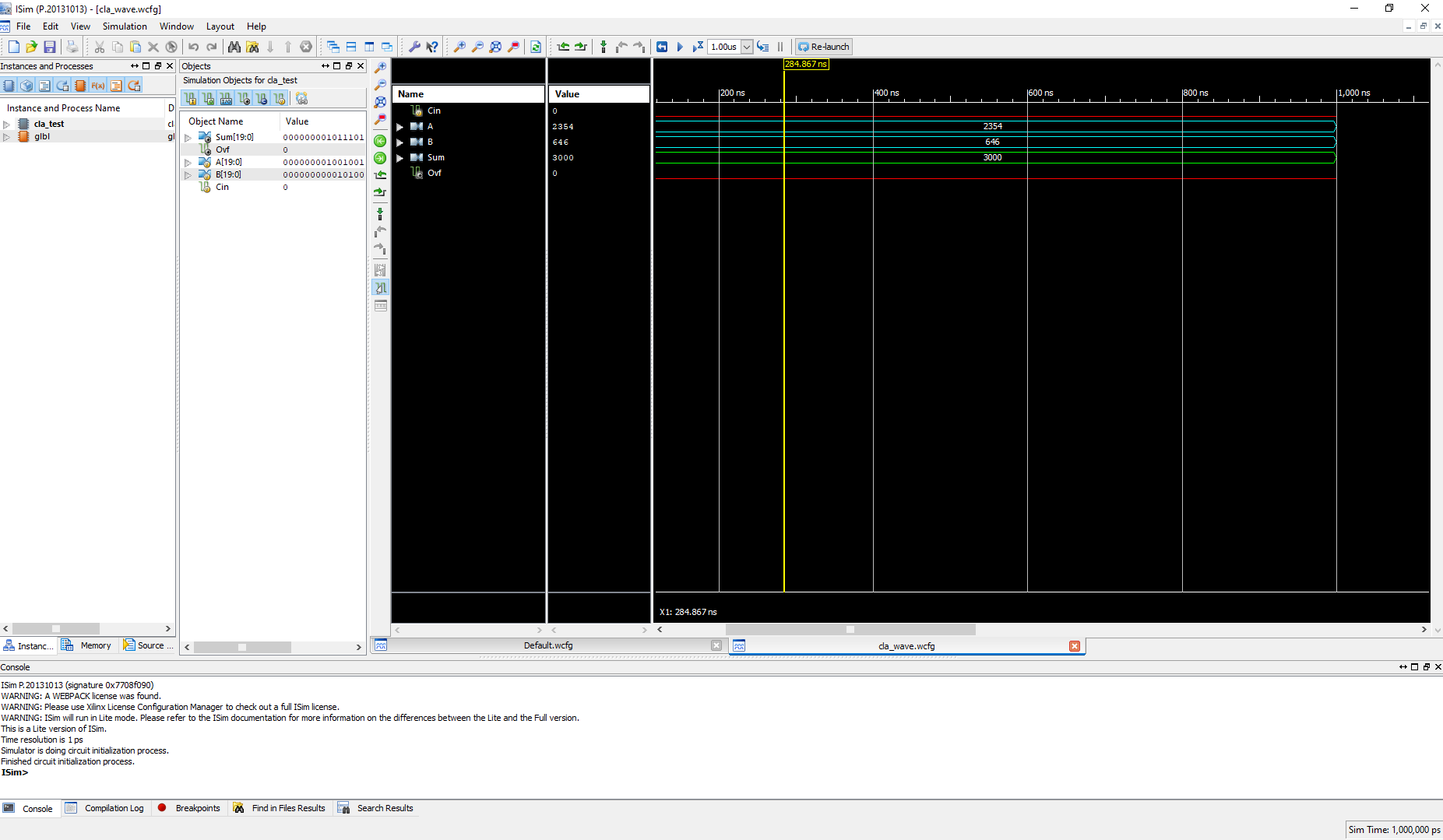
**3.2. Simulation**

In the first image, Cin = 0, A = 524287, B = 1, Sum = -524288 and OVF = 1.

In the second image, Cin = 1, A = 4830, B = 830, Sum = 4000 and OVF = 0.

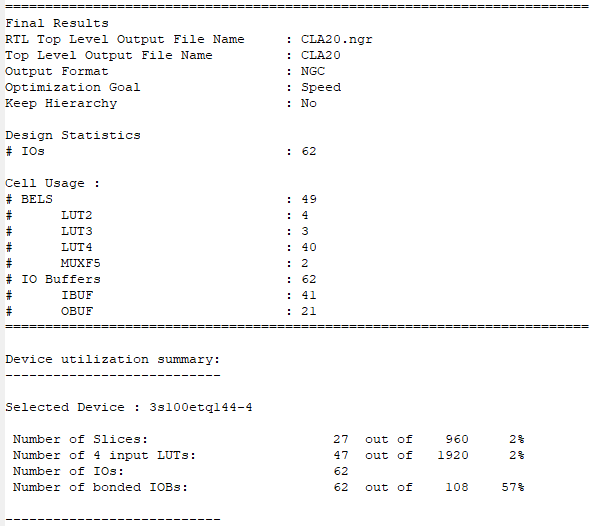


In the third image, Cin = 0, A = -501, B = 401, Sum = -100 and OVF = 0.

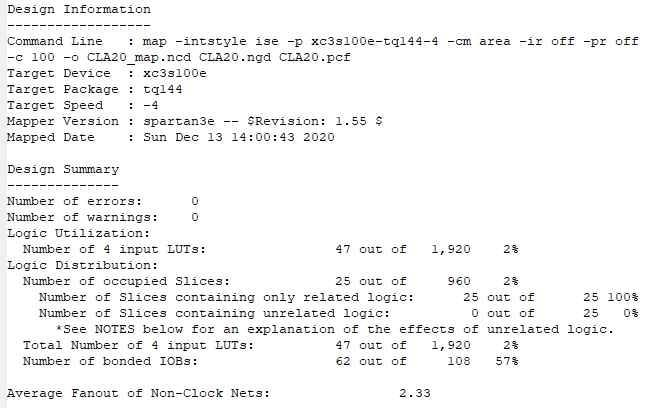
In the fourth image, Cin = 0, A = 2354, B = 646, Sum = 3000 and OVF = 0.

**3.3. Implementation Results**

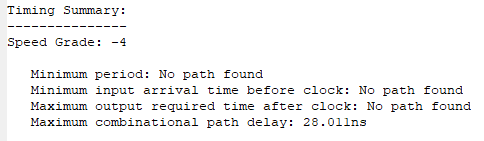
Synthesis report result for 20-bit hybrid adder-subtractor using four 5-bit carry lookahead adder (CLAs) which includes area and timing estimation.

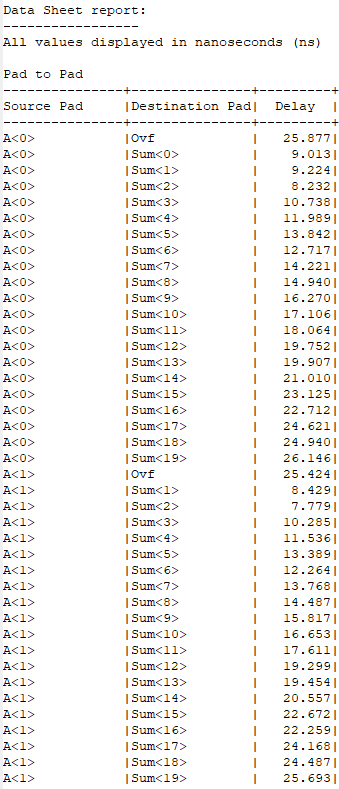
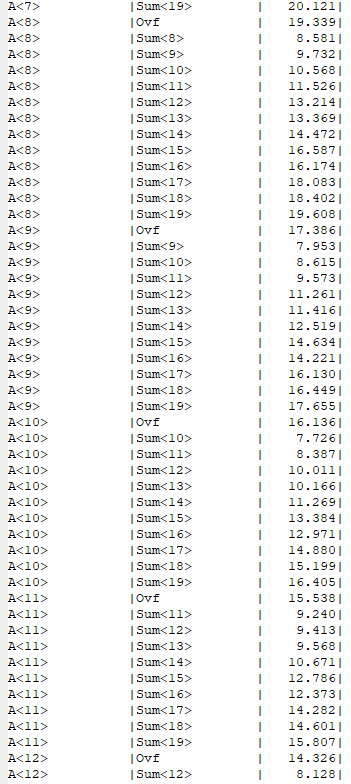
****

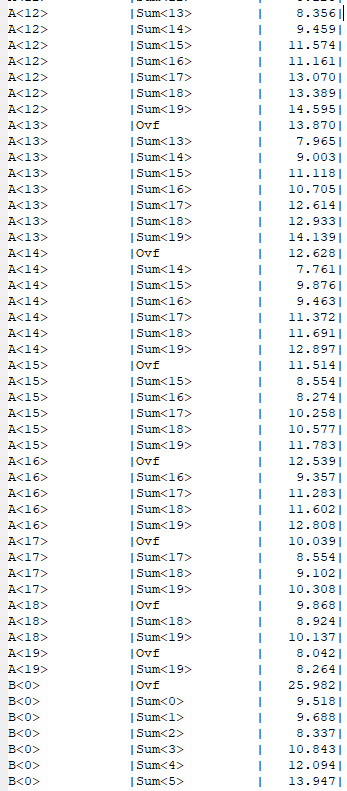
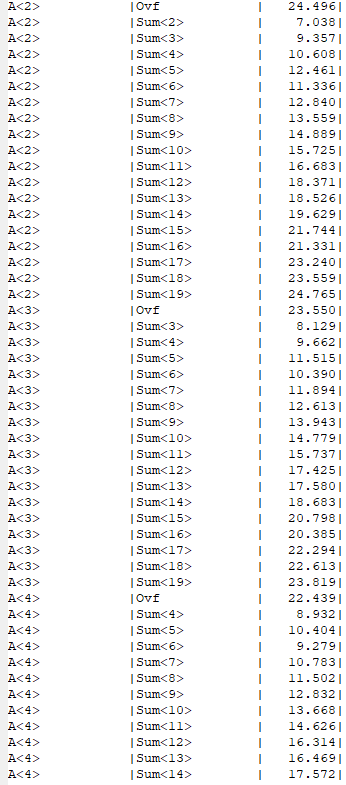
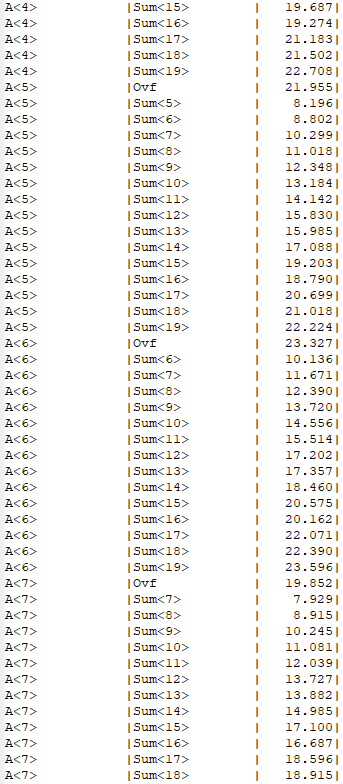
Area for 20-bit hybrid adder-subtractor using four 5-bit carry lookahead adder (CLAs) is 47 (number of 4 input LUTs).

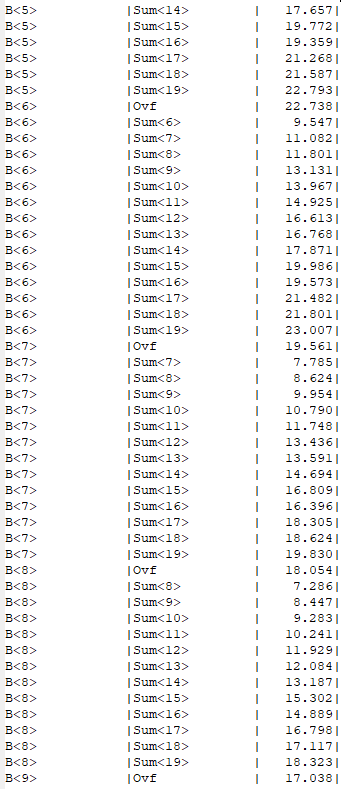
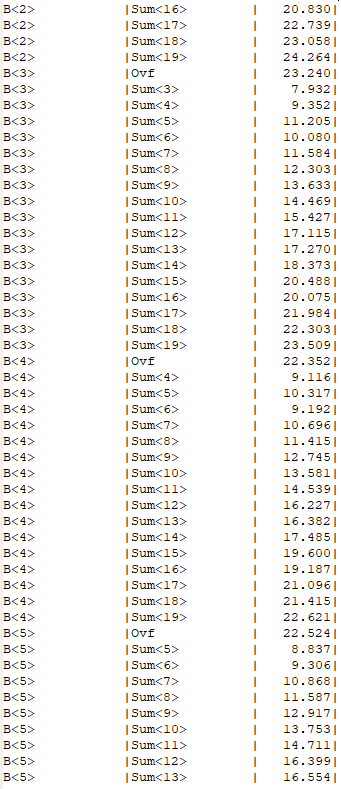
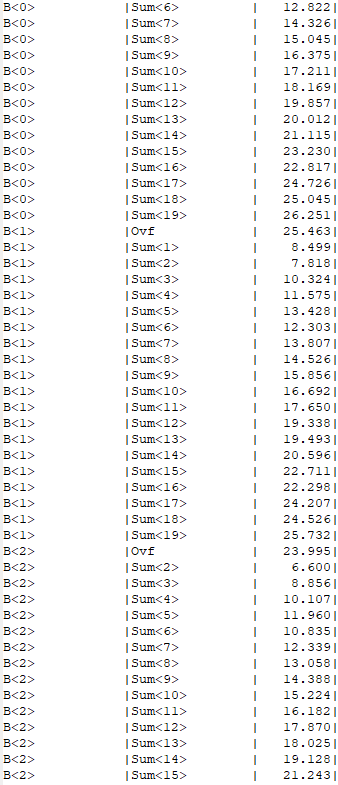


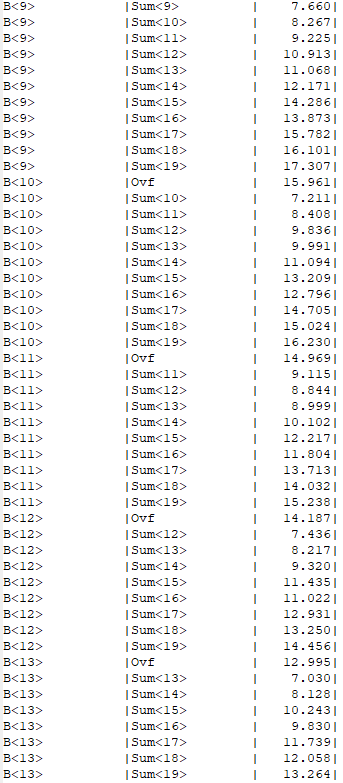
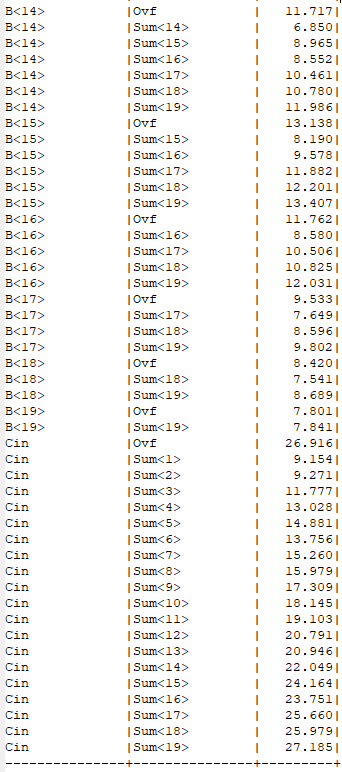
The time for 20-bit hybrid adder-subtractor using four 5-bit carry lookahead adder (CLAs) is 28.011ns (path delay).



Pad to pad for 20-bit hybrid adder-subtractor using four 5-bit carry lookahead adder (CLAs).

****

****

****

1. **Discussion**

**1. For both designs, state synthesis and implementation results. What is the difference between synthesis and implementation results?**

The both syntesis and implementation result are above. For CLA both synthesis and implementation results shows the same number of 4 input LUTs which is 47. Also for ripple carry adder, both synthesis and implementation results shows the same number of 4 input LUTs which is 40.

**2. Which one of the two is better in terms of area (based on implementation result)?**

Area for 20-bit hybrid adder-subtractor using four 5-bit carry lookahead adder (CLAs) is 47

(number of 4 input LUTs). And area for 20-bit ripple-carry adder-subtractor using full adders is 40 (number of 4 input LUTs).

So, we can say that 20-bit ripple-carry adder-subtractor using full adders is better than 20-bit hybrid adder-subtractor using four 5-bit carry lookahead adder (CLAs) in terms of area because it is using less area than CLA.

**3. Which one of the two is better in terms of time (based on implementation result)?**

The time for 20-bit hybrid adder-subtractor using four 5-bit carry lookahead adder (CLAs) is 28.011ns (path delay). And the time for 20-bit ripple-carry adder-subtractor using full adders is 28.977ns (path delay).

So, we can say that 20-bit hybrid adder-subtractor using four 5-bit carry lookahead adder (CLAs) is better than 20-bit ripple-carry adder-subtractor using full adders in terms of time because CLA is working faster than ripple carry adder.

**4. Define a new metric to measure the time-area tradeoff in two designs by multiplying the**

**number of LUTs and time. Which one of the two designs is better in terms of this new**

**metric?**

For 20-bit ripple-carry adder-subtractor using full adders; area = 40 and time = 28.977ns

With new metric system = 1159.08 number of 4 input LUTs/ns

For 20-bit hybrid adder-subtractor using four 5-bit carry lookahead adder (CLAs); area = 47, time = 28.011ns, with new metric system = 1316.517 number of 4 input LUTs/ns

20-bit ripple carry adder is the better design because it has a smaller number of 4 input LUTs/ns than the 20-bit hybrid adder-subtractor using four 5-bit carry lookahead adder (CLAs). Also, it is better around 1.13 multiple. When the new metric system is smaller, it shows that time and area is more efficient than the other one.