Implementation and Analyzing 64-bit Carry Select Adders

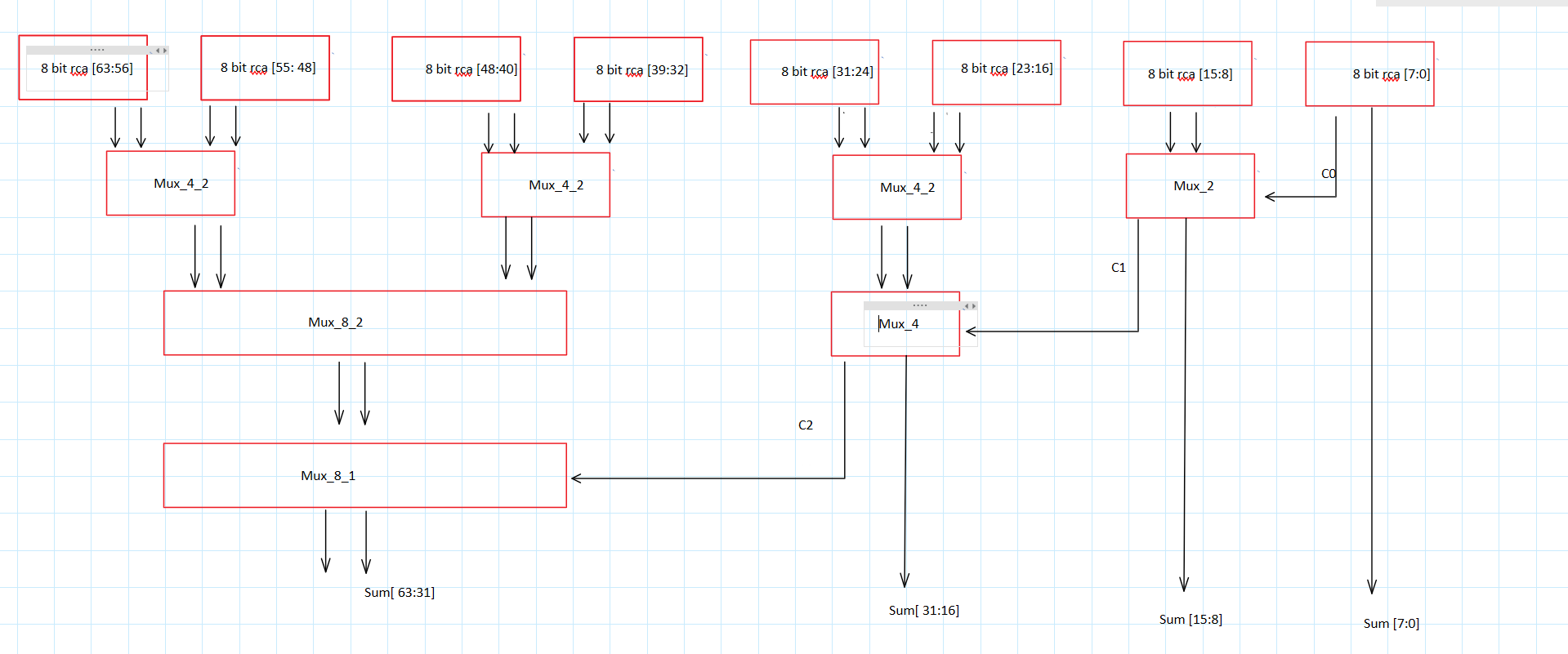
* Eakta Anchlia

The intension of the project to implement several types of 64-bit Carry Sum Adder adders and performing detailed analysis of area, timing and power.

**Carry Sum Adder** is implemented using 8 instances 8-bit ripple carry adder, which is made using 8 one-bit adders. The first bin is calculated using half\_adder in the first ripple carry as there is no cin. Below is the list of ports and description

|  |  |  |
| --- | --- | --- |
| **Port** | **Input/Output** | **Short Description** |
| op1[63:0] | Input | Operand 1 |
| op2 [63:0] | Input | Operand 2 |
| Addsum | Input | 0 : Addition  1: Subtraction |
| Clock | Input | Clock to the design |
| Overflow | Input | Bit signifies the sign of the sum |
| Start | Input | A positive edge pulse to load design qith operands |
| Reset | Input | Resets Values |
| sum [63:0] | Output | Result store in sum |
| cout | Output | Carry out result in cout |

**Figure 1**: Overall Block Diagram of the 64-bit Carry Sum



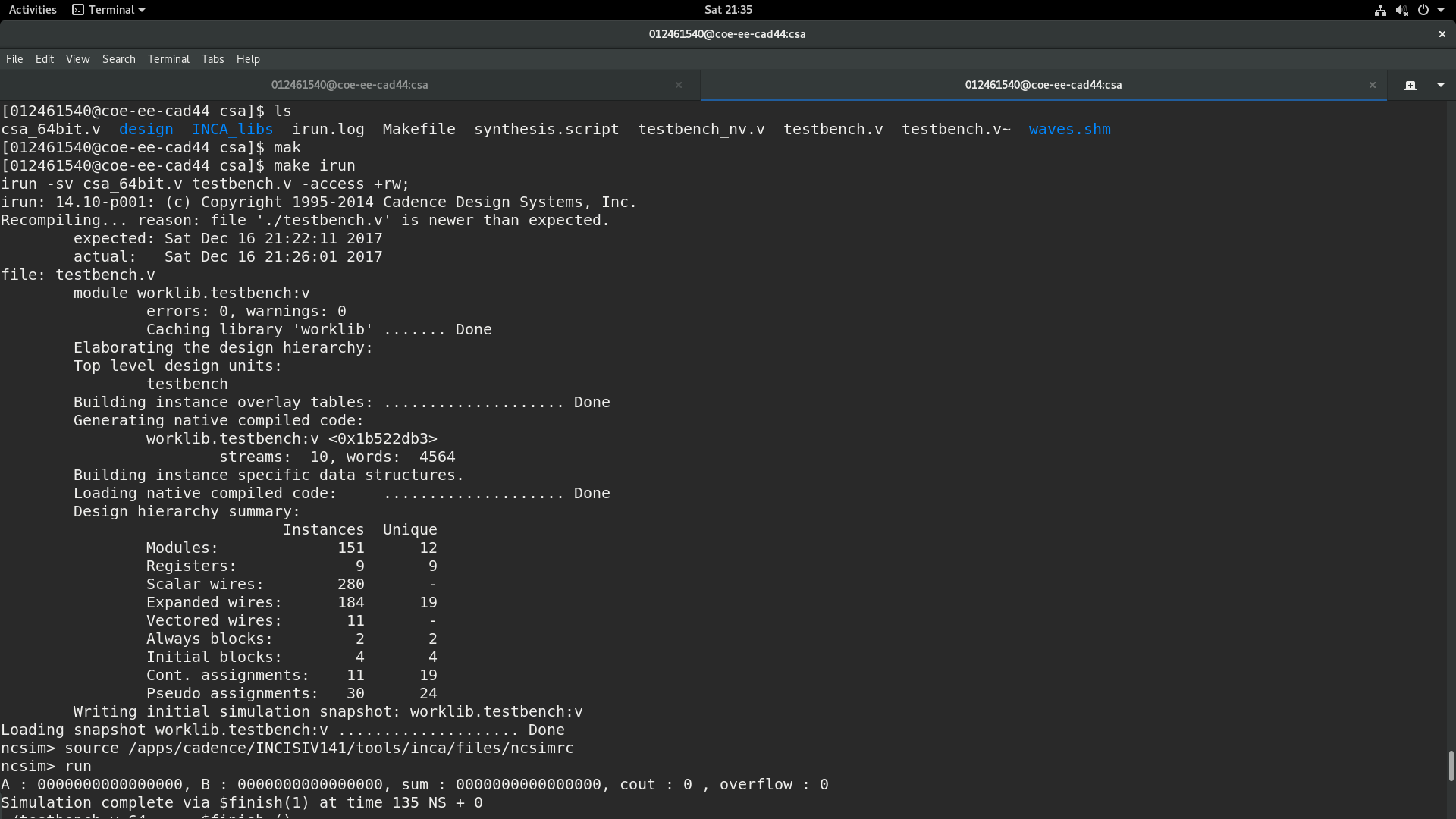
**RTL-Level (Pre-synthesis) Simulations/Tests**

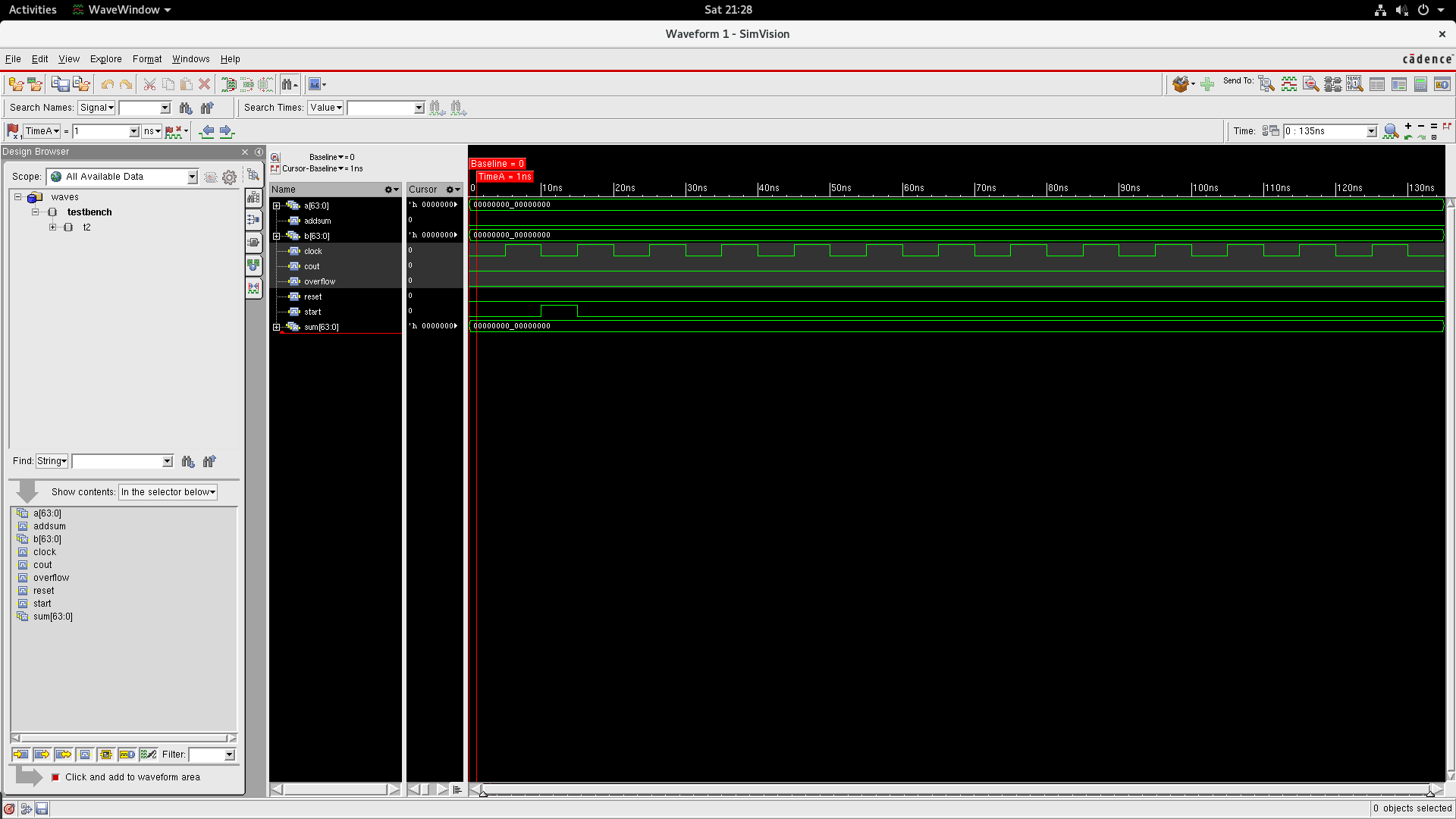
The different 5 test-benches are used with 6 different pair of inputs to verify the design. To verify the functionality of reset, the reset is also given in between the transitions of the inputs. 6 different inputs are selected such that major coverage of the design can be achieved. One of the inputs is all 0, this input is chosen to verify that none of the gate is misbehaving and giving all output as 0 without any glitches. Other pairs of inputs are such that ripple carry from one block to another can be verified.

**Table 1** – Four Selected Test Data for Multiplier Circuit

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Testcase | Operand1 (hex) | Operand1 (hex) | Addsum | cout | Sum (hex) | Overflow |
| 1 | 0000\_0000\_0000\_0000 | 0000\_0000\_0000\_0000 | 0 | 0 | 0000\_0000\_0000\_0000 | 0 |
| 2 | ffff\_ffff\_ffff\_ffff | 0000\_0000\_0000\_0000 | 0 | 0 | 0000\_0000\_0000\_0000 | 0 |
| 3 | ffff\_ffff\_ffff\_ffff | 0000\_0000\_0000\_0000 | 1 | 1 | 0000\_0000\_0000\_0000 | 0 |
| 4 | 14ab78efd8535c7d | 8dfdbc24bd8f18c0 | 0 | 0 | 86adbccb1ac443bd | 1 |
| 5 | 1229801703532086340 | 0000\_0000\_0000\_0001 | 0 | 1 | 1229801703532086339 | 0 |

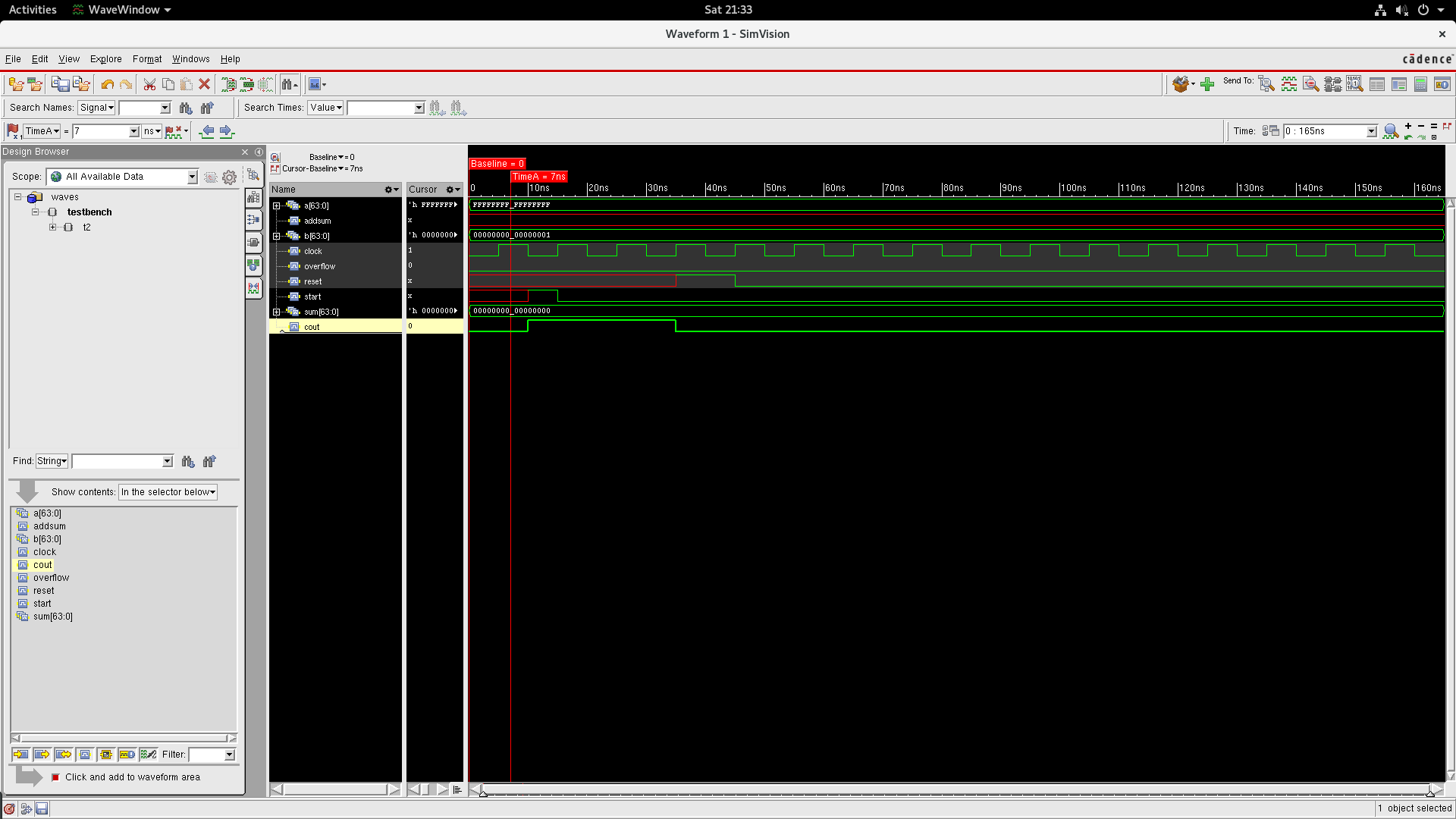
**Figure 2**: RTL simulation waveform that contains test case #1



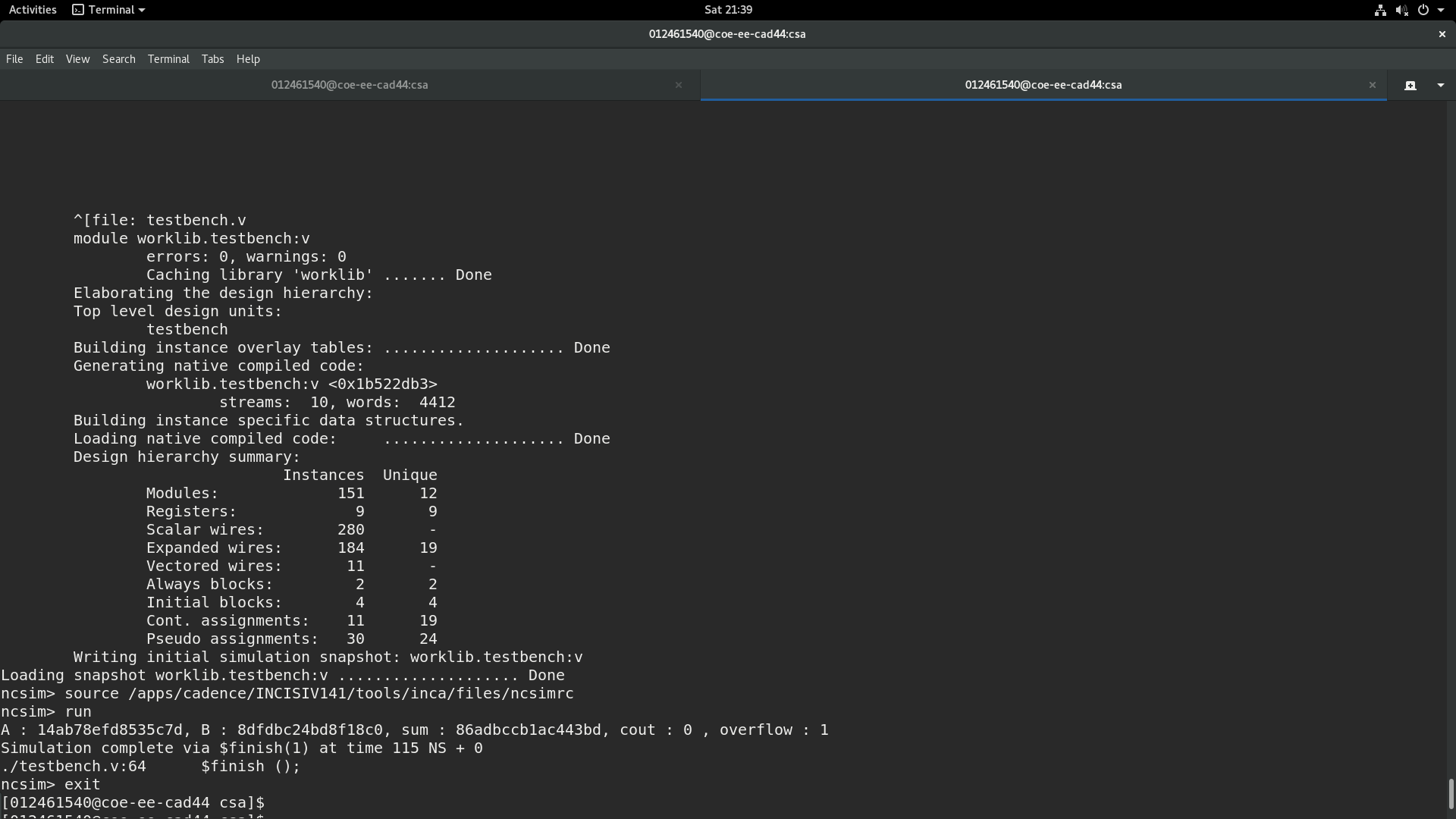


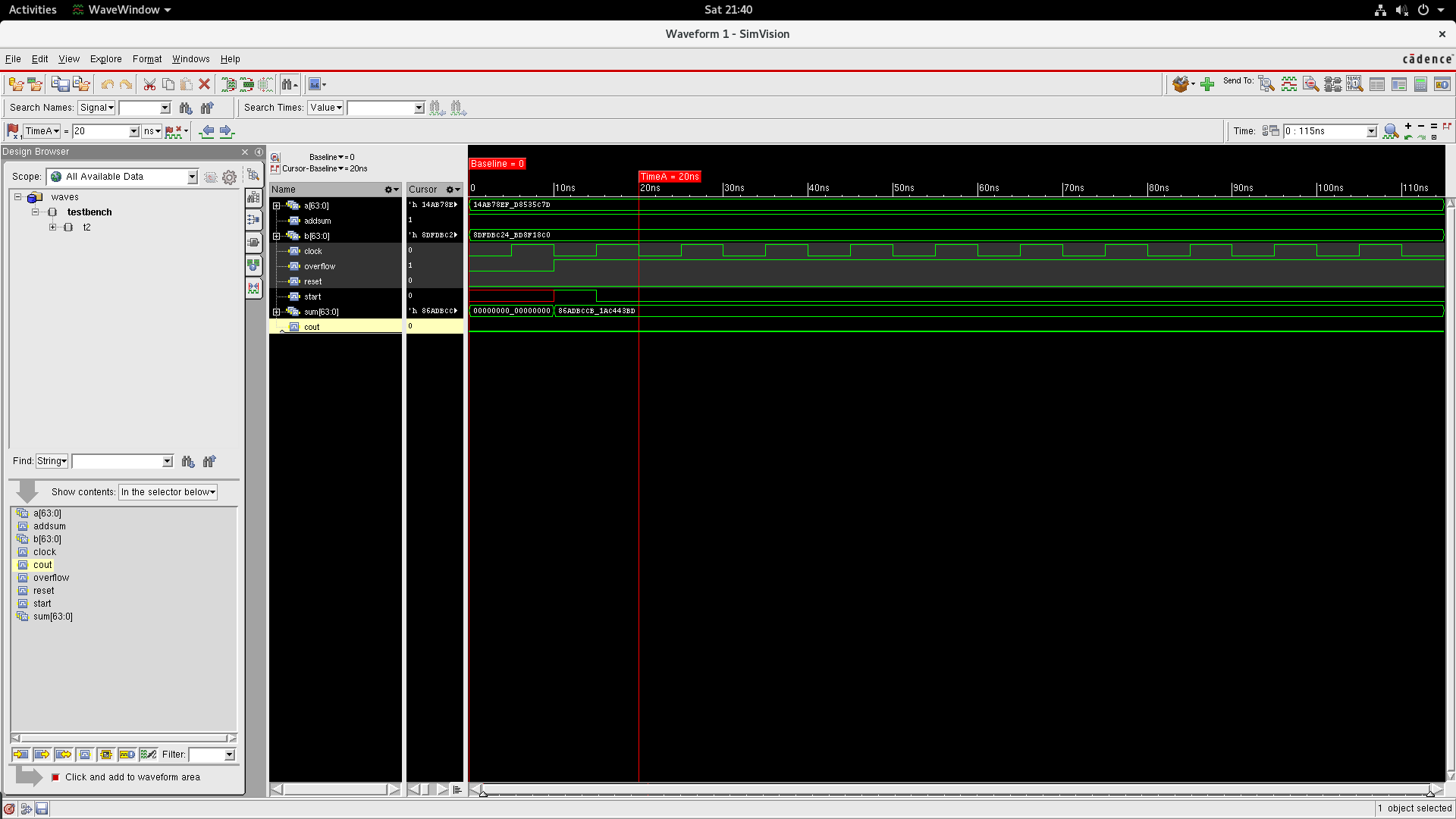
**Figure 3**: RTL simulation waveform that contains test case #2





**Figure 4**: RTL simulation waveform that contains test case #3





**Conclusion**

Carry sum adder :

Area : 2027.500

Total Negative Slack : 0

Total Dynamic Power : 6.0314 mW

In conclusion:

1. More the number of hardware, faster is the operation of circuit.
2. Area is Inversely proportional to delay.
3. Area is Directly proportional to Power of the circuit.
4. Time required by the circuit for stable output depends on the given test case values.