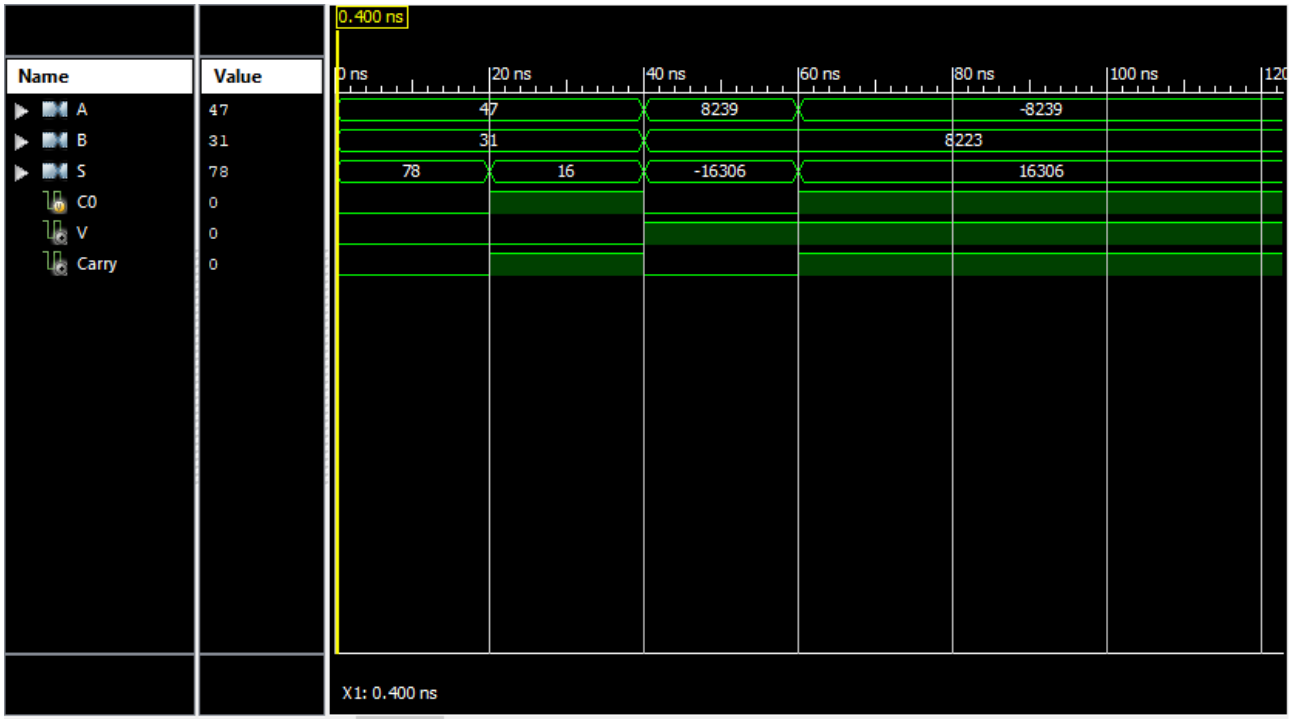
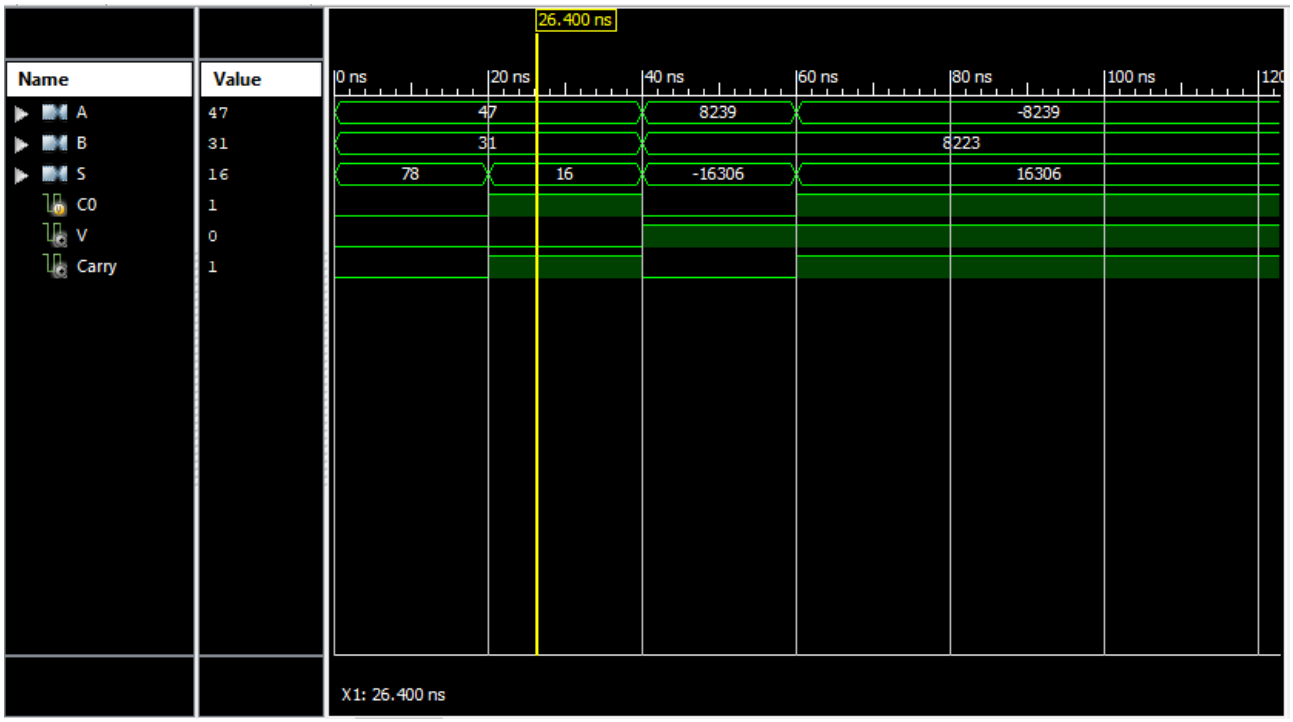
1. 15-bit carry-ripple adder-subtractor using full adders

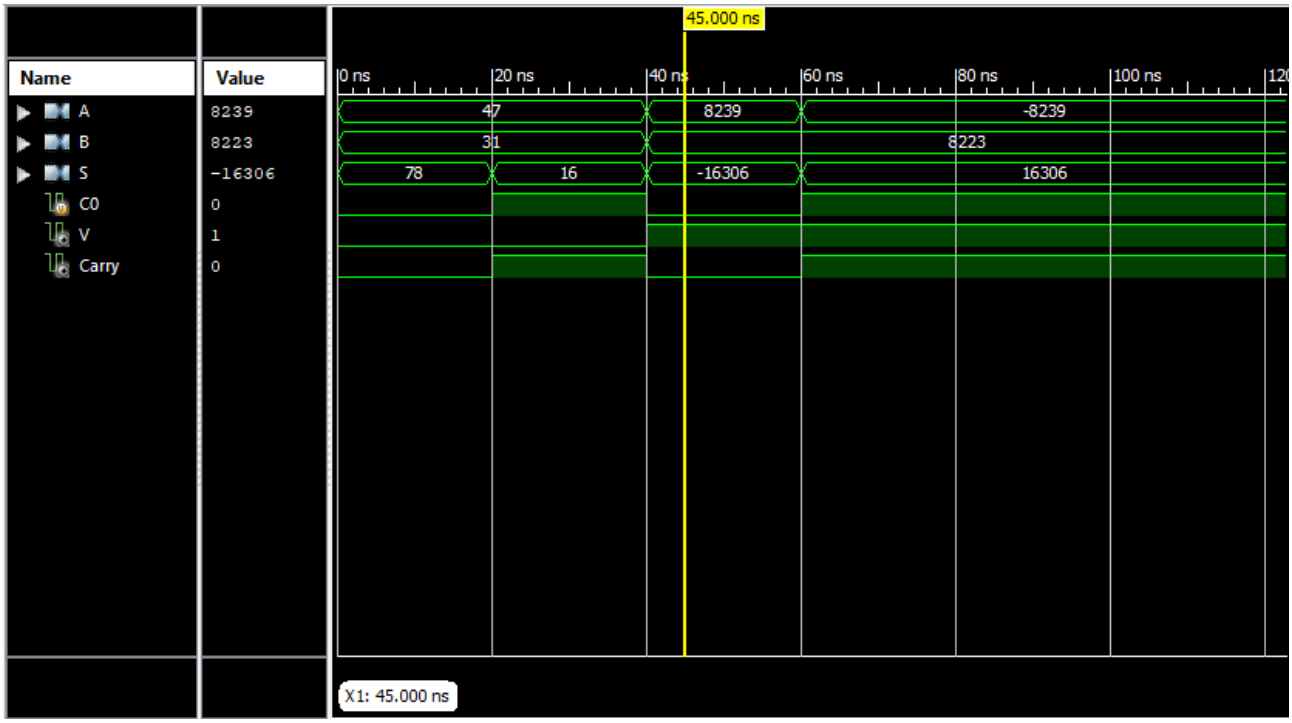
Addition with no overflow



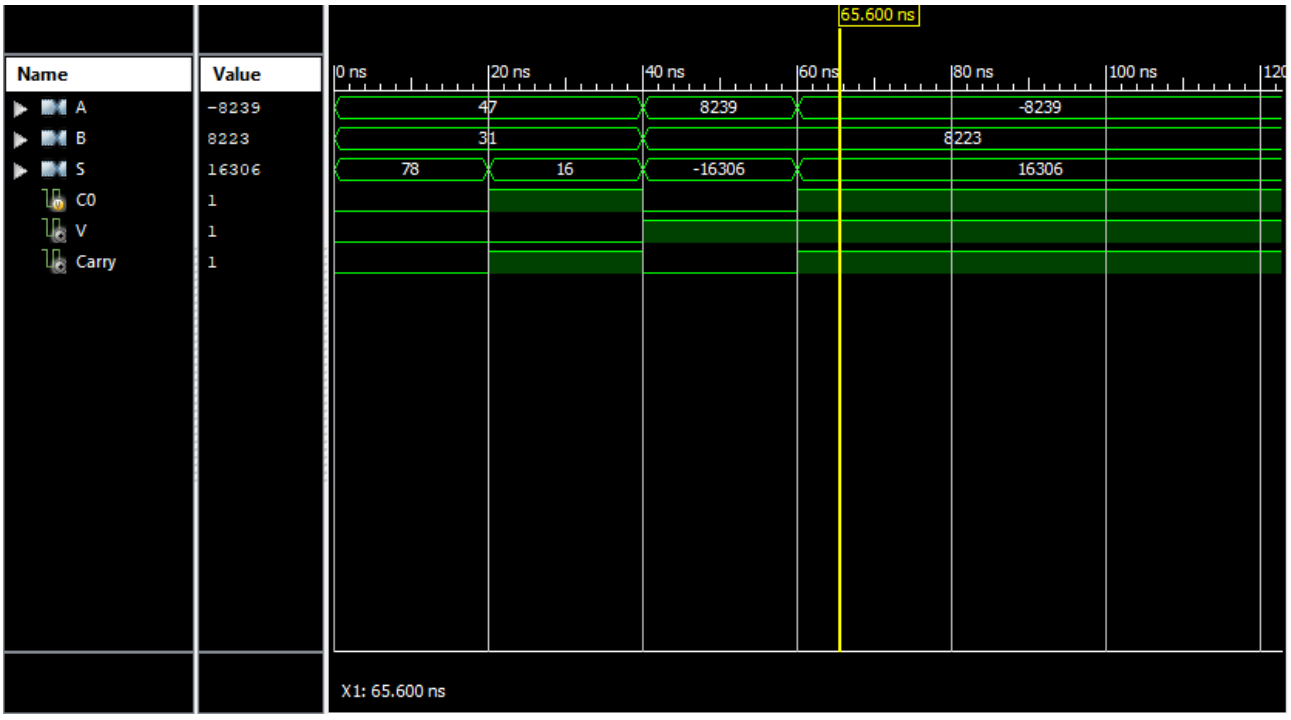
Subtraction with no overflow



Addition with overflow



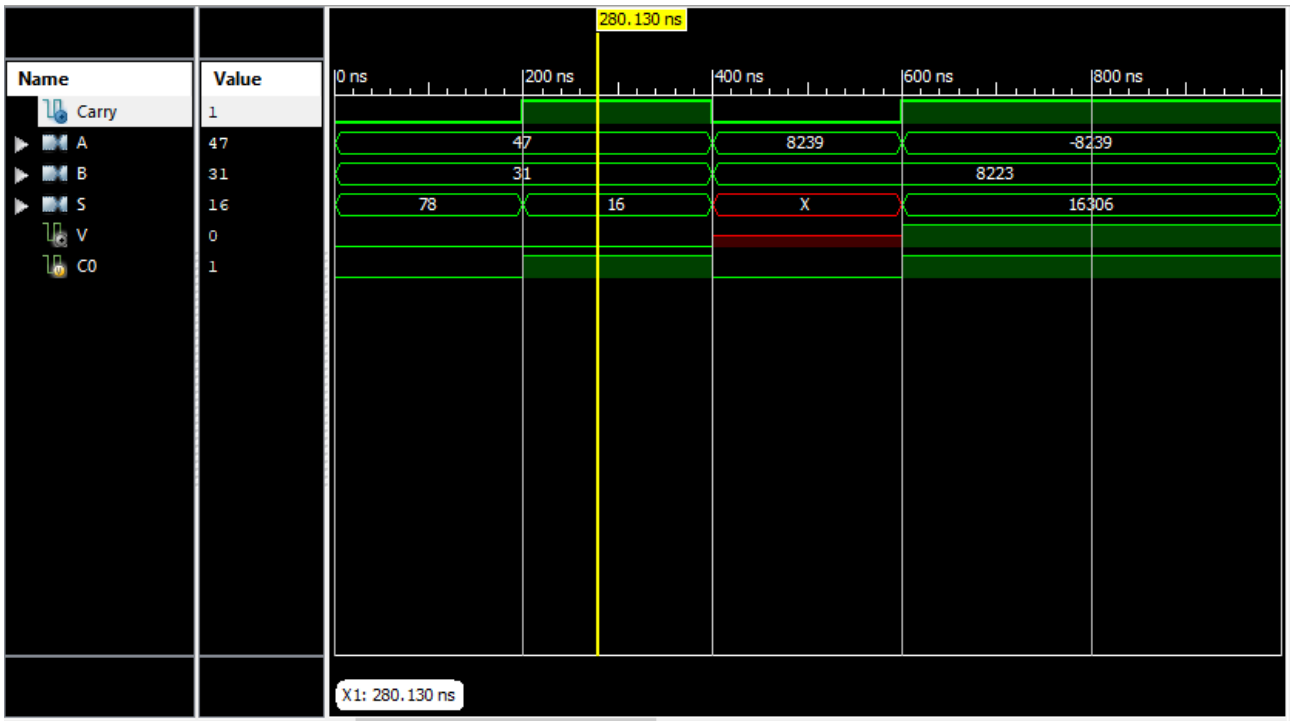
Subtraction with overflow



2. 15-bit hybrid adder-subtractor using a number of carry lookaheads

Addition with no overflow

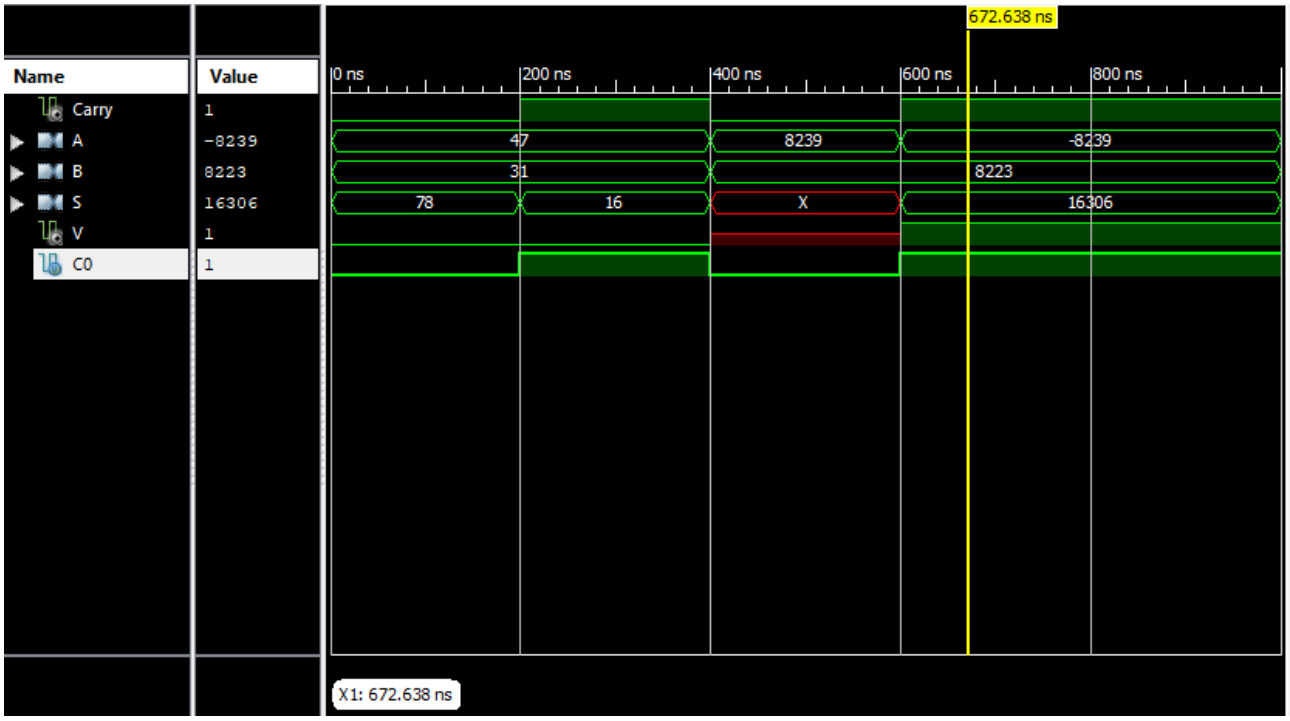
Subtraction with no overflow



Addition with overflow

We tried everything but we couldn't solve the 'red area' problem. Other than that everything works fine.

Subtraction with overflow



1. Which one of the two is better in terms of area?

Carry-ripple adder-subtractor is better in terms of area because lookahead's LUT value is 88 and carry-ripple adder-subtractor's LUT value is 59, so we can easily see that carry-ripple adder-subtractor is better in terms of area.

2. Which one of the two is better in terms of time?

Hybrid adder-subtractor is better in terms of area because lookahead's time is 21.122ns and carry-ripple adder-subtractor's time is 43.176ns, so we can easily see that Hybrid adder-subtractor is better in terms of time.

3. 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?

We declared our metric as volume = LUT x time, so when volume gets big it means that we used a lot of space or time or both, so when we calculate this in our adder-subtractors,
Carry-ripple : 59 x 43.176 = 2547.384 and Hybrid : 88 x 21.122 = 1858.736
So Hybrid is better in terms of our new metric.

4. State the requirements of a *good* design in terms of area, time and the new metric you've defined.

In order to get good design our time and area should be low.