CS303 – Logic & Digital System Design CS303 – Lab#3 Report

Name-Surname/SID: Jeren Annagurbanova/28517

1. Project Description

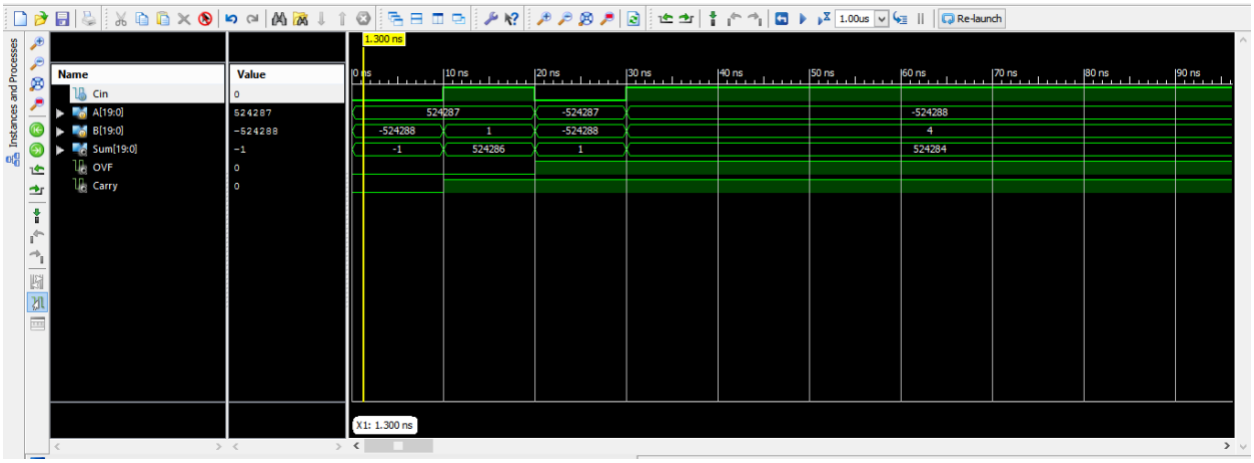
Designing a circuit that can both add and subtract two signed 20-bit integers and realize whether there is an overflow or not.

Design:

1. 20-bit ripple-carry adder-subtractor using full adders

2. 20-bit hybrid adder-subtractor using five 5-bit carry lookahead adders (CLAs). Both adder-subtractors must detect overflow

2. First Design



4 different test cases with addition and subtraction, both with and without overflow:

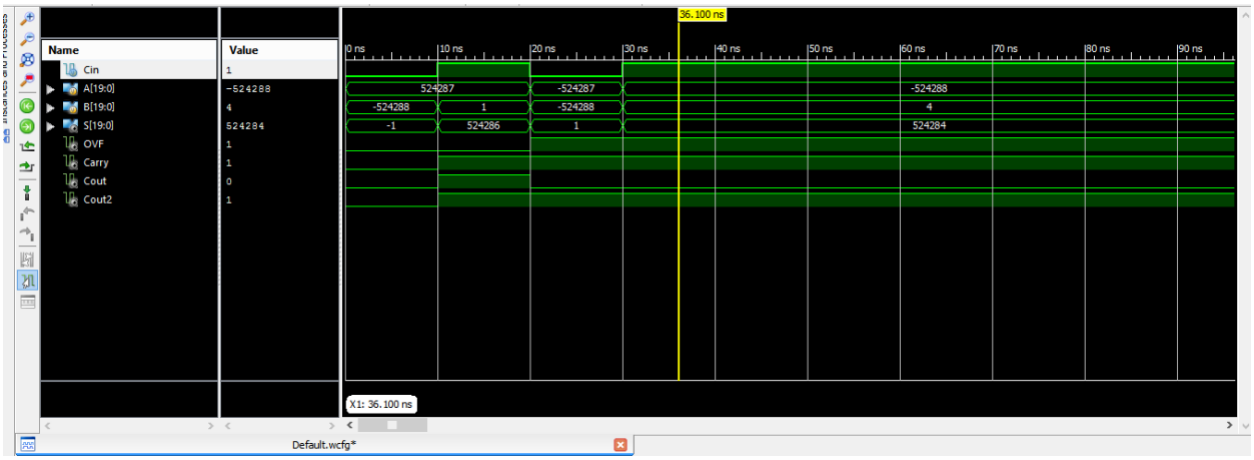
```
initial begin
   // Initialize Inputs
   //Addition with no oveflow (524287)+ (-524288) = should give -1
   A = (524287);
   B = (-524288);
   Cin = (0);
   #10;
   //subtraction with no overflow (524287) - 1 should give 524286
   A = (524287);
   B = (1);
   Cin = (1);
   #10;
   //Addition with overflow (-524287) + (-524288) should give overflow
   A = (-524287);
   B = (-524288);
   Cin = (0);
   #10;
   //Subtraction with Overflow (-524288) - (4) should give overflow
   A = (-524288);
   B = (4);
   Cin = (1);
   #10;
```

Implementation Results Present implementation results

Area: # of LUTS 41 out of 1920

Time: 29.018ns

3. Second Design



4 different test cases with addition and subtraction, both with and without overflow:

```
52      // Initialize Inputs
53      //Addition with no oveflow (524287)+ (-524288) = should give -1
54      A = (524287);
55      B = (-524288);
56      Cin = (0);
57      #10;
58      //subtraction with no overflow (524287) - 1 should give 524286
59      A = (524287);
60      B = (1);
61      Cin = (1);
62      #10;
63      //Addition with overflow (-524287) + (-524288) should give overflow
64      A = (-524287);
65      B = (-524288);
66      Cin = (0);
67      #10;
68      //Subtraction with Overflow (-524288) - (4) should give overflow
69      A = (-524288);
70      B = (4);
71      Cin = (1);
72      #10;
73
```

Implementation Results Present implementation results

Area: # of LUTS 48 out of 1920

Time: 28.042ns

4. Questions

1) Ripple Carry:
   Area: # of LUTS 41 out of 1920; Time: 29.018ns

   Carry Lookahead:
   Area: # of LUTS 48 out of 1920； Time: 28.042ns

   Difference in terms of area: |48 – 41| = 7
   Difference in terms of time: |28.042 - 29.018| = 0.976 ns

2) Ripple carry adder/subtractor is better in terms of Area, its area is less than CL's area, (41 < 48)

3) CLA/S is better in terms of Time, it is faster than Ripple carry
   (28.042ns < 29.018ns)

4) A new metric to measure the time-area tradeoff in two designs by multiplying the number of LUTs and time.

   For RC: 41* 29.018ns = 1189.738
   For CL: 48 * 28.042ns = 1346.016

In this case, Ripple carry adder/subtractor is better according to the new metric