Lab #3
CS303

Deniz Küçükahmetler

24879

December 16, 2020

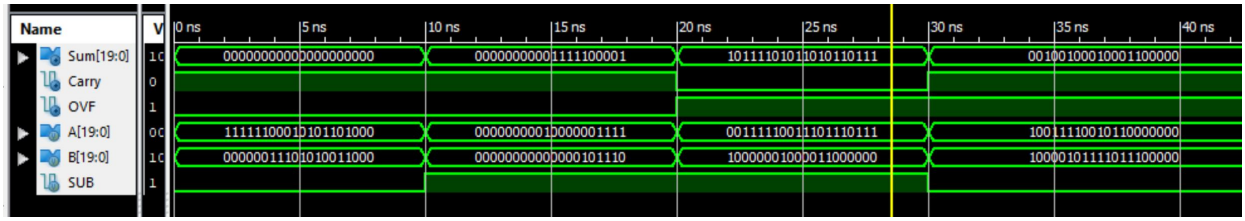
1 Project Description

Designing a 15-bit adder/subractor.
In the first design, a 20-bit ripple-carry adder-subtractor using full adders was designed.
In the second design a 20-bit hybrid adder-subtractor using four 5-bit carry lookahead adders was designed.

2 First Design

2.1 Simulation

For simulation, 4 different inputs were tried. Bit representations of inputs, outputs, adder/subtractor indicator input (SUB = 0 for addition, SUB = 1 for subtraction), overflow detection output (OVF = 0 for no overflow, OVF = 1 for overflow) can be seen in the figure.



Here is the input/output flow by decimal representations:

- input A: -15000
  input B: 15000
  operation: addition
  result: 0
  overflow: no

- input A: 1039
  input B: 46
  operation: subtraction
  result: 993
  overflow: no

- input A: 155863
  input B: -520000
  operation: subtraction
  result: 272713
  overflow: yes

- input A: -400000
  input B: -500000
  operation: addition
  result: 148528
  overflow: yes

2.2 Synthesis Results

Area:

```
=========================================================================
*                            Final Report
=========================================================================
Final Results
RTL Top Level Output File Name     : RCA.ngr
Top Level Output File Name         : RCA
Output Format                      : NGC
Optimization Goal                  : Speed
Keep Hierarchy                     : No

Design Statistics
# IOs                              : 63

Cell Usage :
# BELS                             : 41
#      LUT2                        : 1
#      LUT3                        : 1
#      LUT4                        : 39
# IO Buffers                       : 63
#      IBUF                        : 41
#      OBUF                        : 22
=========================================================================

Device utilization summary:
---------------------------

Selected Device : 3s100etq144-4

 Number of Slices:                       24  out of     960     2%
 Number of 4 input LUTs:                 41  out of    1920     2%
 Number of IOs:                          63
 Number of bonded IOBs:                  63  out of     108    58%
```

It can be seen in the figure that, Number of LUTs: 41 out of 1,920.
Time:

```
Timing Summary:
---------------
Speed Grade: -4

   Minimum period: No path found
   Minimum input arrival time before clock: No path found
   Maximum output required time after clock: No path found
   Maximum combinational path delay: 29.158ns
```

So, maximum combinational combinational path delay is 29,158 ns.

2.3 Implementation Results

Area:

```
Design Summary
--------------
Number of errors:      0
Number of warnings:    0
Logic Utilization:
  Number of 4 input LUTs:                41 out of   1,920    2%
Logic Distribution:
  Number of occupied Slices:             21 out of     960    2%
    Number of Slices containing only related logic:      21 out of      21 100%
    Number of Slices containing unrelated logic:          0 out of      21   0%
      *See NOTES below for an explanation of the effects of unrelated logic.
  Total Number of 4 input LUTs:          41 out of   1,920    2%
  Number of bonded IOBs:                 63 out of     108   58%
```

It can be seen in the figure that, Number of 4 input LUTs: 41 out of 1,920.
Time:

```
B<18>          |Sum<18>         |    7.052|
B<18>          |Sum<19>         |    8.280|
B<19>          |Carry           |    6.810|
B<19>          |OVF             |    7.720|
B<19>          |Sum<19>         |    8.152|
Cin            |Carry           |   26.858|
Cin            |OVF             |   27.324|
Cin            |Sum<1>          |    9.681|
Cin            |Sum<2>          |   11.093|
Cin            |Sum<3>          |   12.761|
Cin            |Sum<4>          |   13.210|
Cin            |Sum<5>          |   14.079|
Cin            |Sum<6>          |   15.572|
Cin            |Sum<7>          |   16.066|
Cin            |Sum<8>          |   17.179|
Cin            |Sum<9>          |   18.300|
Cin            |Sum<10>         |   19.585|
Cin            |Sum<11>         |   19.681|
Cin            |Sum<12>         |   20.687|
Cin            |Sum<13>         |   22.518|
Cin            |Sum<14>         |   21.865|
Cin            |Sum<15>         |   23.975|
Cin            |Sum<16>         |   24.953|
Cin            |Sum<17>         |   26.287|
Cin            |Sum<18>         |   26.349|
Cin            |Sum<19>         |   27.582|
---------------+----------------+---------+


Analysis completed Wed Dec 16 12:00:40 2020
--------------------------------------------------------------------------------
```

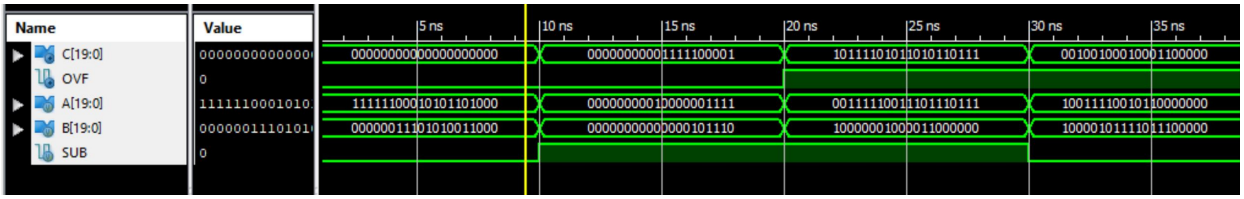
So, maximum delay received in the design is 27.582 ns.

3 Second Design

In this design, a hybrid 20-bit carry look-around adder/subtractor was designed.

3.1 Simulation

For simulation, 4 different inputs were tried. Bit representations of inputs, outputs, adder/subtractor indicator input (SUB = 0 for addition, SUB = 1 for subtraction), overflow detection output (OVF = 0 for no overflow, OVF = 1 for overflow) can be seen in the figure.



Here is the input/output flow by decimal representations:

- input A: -15000
  input B: 15000

operation: addition
result: 0
overflow: no

- input A: 1039
  input B: 46
  operation: subtraction
  result: 993
  overflow: no

- input A: 155863
  input B: -520000
  operation: subtraction
  result: 272713
  overflow: yes

- input A: -400000
  input B: -500000
  operation: addition
  result: 148528
  overflow: yes

3.2 Synthesis Results

Area:

```
=========================================================================
*                           Final Report                                *
=========================================================================
Final Results
RTL Top Level Output File Name     : cla_add_sub.ngr
Top Level Output File Name         : cla_add_sub
Output Format                      : NGC
Optimization Goal                  : Speed
Keep Hierarchy                     : No

Design Statistics
# IOs                              : 62

Cell Usage :
# BELS                             : 49
#      LUT2                        : 4
#      LUT3                        : 3
#      LUT4                        : 40
#      MUXF5                       : 2
# IO Buffers                       : 62
#      IBUF                        : 41
#      OBUF                        : 21
=========================================================================

Device utilization summary:
---------------------------

Selected Device : 3s100etq144-4

 Number of Slices:                       27  out of    960     2%
 Number of 4 input LUTs:                 47  out of   1920     2%
 Number of IOs:                          62
 Number of bonded IOBs:                  62  out of    108    57%
```

It can be seen that, number of 4 input LUTs is 47 out of 1.920.
Time:

```
Timing Summary:
---------------
Speed Grade: -4

   Minimum period: No path found
   Minimum input arrival time before clock: No path found
   Maximum output required time after clock: No path found
   Maximum combinational path delay: 27.949ns
```

It can be seen that, maximum combinational path delay of the design is 27,949 ns.

3.3 Implementation Results

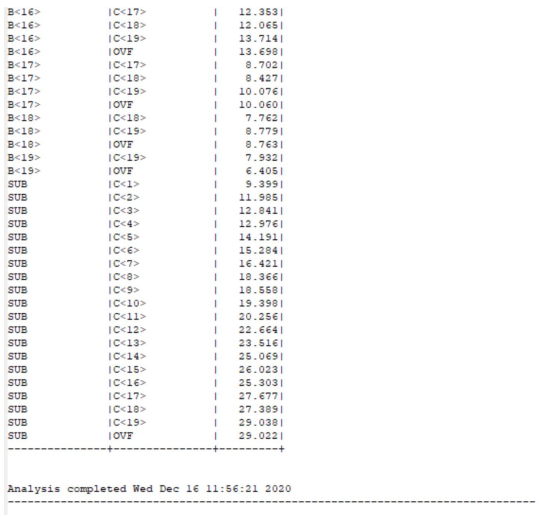
Area:

```
Design Summary
--------------
Number of errors:      0
Number of warnings:    0
Logic Utilization:
  Number of 4 input LUTs:                47 out of   1,920    2%
Logic Distribution:
  Number of occupied Slices:             26 out of     960    2%
    Number of Slices containing only related logic:      26 out of      26 100%
    Number of Slices containing unrelated logic:          0 out of      26   0%
      *See NOTES below for an explanation of the effects of unrelated logic.
  Total Number of 4 input LUTs:          47 out of   1,920    2%
  Number of bonded IOBs:                 62 out of     108   57%

Average Fanout of Non-Clock Nets:                2.33

Peak Memory Usage:  4382 MB
Total REAL time to MAP completion:  1 secs
Total CPU time to MAP completion:   1 secs
```

It can be seen that, number of 4 input LUTs is 47 out of 1.920.
Time:



B<16>	C<17>	12.353
B<16>	C<18>	12.065
B<16>	C<19>	13.714
B<16>	OVF	13.698
B<17>	C<17>	8.702
B<17>	C<18>	8.427
B<17>	C<19>	10.076
B<17>	OVF	10.060
B<18>	C<18>	7.762
B<18>	C<19>	8.779
B<18>	OVF	8.763
B<19>	C<19>	7.932
B<19>	OVF	6.405
SUB	C<1>	9.399
SUB	C<2>	11.985
SUB	C<3>	12.841
SUB	C<4>	12.976
SUB	C<5>	14.191
SUB	C<6>	15.284
SUB	C<7>	16.421
SUB	C<8>	18.366
SUB	C<9>	18.558
SUB	C<10>	19.398
SUB	C<11>	20.256
SUB	C<12>	22.664
SUB	C<13>	23.516
SUB	C<14>	25.069
SUB	C<15>	26.023
SUB	C<16>	25.303
SUB	C<17>	27.677
SUB	C<18>	27.389
SUB	C<19>	29.038
SUB	OVF	29.022

Analysis completed Wed Dec 16 11:56:21 2020

It can be seen that, maximum delay of the design is 29,022 ns.

4 Discussion

- Ripple-carry adder/subtractor gave 29,158 ns of delay whereas CLA gave 27,949 ns of delay in synthesis results.
  Ripple-carry adder/subtractor gave 27,582 ns of delay whereas CLA gave 29,022 ns of delay in implementation results.
  In both synthesis and implementation results, 41 LUTs were given for ripple-carry and 47 LUTs were given for CLA.
  In synthesis results, ripple-carry adder/subtractor was better in terms of area and worse in terms of speed compared to CLA. Thus, the area-speed trade-off can be observed. However in implementation, ripple-carry adder was better in terms of both area and speed. This is theoretically wrong. So, this situation can be explained with outer factors such as computers operating system or hardware properties.

- First design was better in terms of time when we compare the maximum delay times. Again, first design was better in terms of area. However, this wasn't expected as stated above and can be explained with outer factors, perhaps depending on the computer hardware.

- A metric as time x area can be defined. Since we want both of them to be small, we can say that if the result of time x area is less, the design is better. Time is measured in ns and area is measured in terms of LUTs. First design = 58 x 25.210 = 1462.18 LUTs x ns Second design = 85 x 20.698 = 1759.33 LUTs x ns First design is better in terms of the new metric defined.

- In a good design, it is desired to occupy less area and and short time. In the metric I defined, both time and space are taken into account. If one is too much, metric grows more since time and area are multiplied. Of course a "good design" depends on the purpose of the task. A design can be preferred for a specific task but not for others.

In the designs we made, there is no better design since ripple-carry is better in terms of area and CLA is better in terms of speed. So, better design depends on the purpose and the bit length of the inputs.