Graehme Blair

Drew Hasse

EECS 355

Lab 2: Combinational Divider

**Simulation Results**

Three simulations were run. Two divider test (32bit/16bit) and one comparator test (4bit).

The test cases for the comparator were hard coded.

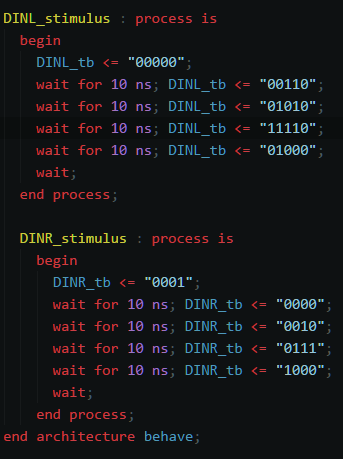
Numbers for the divider tests were read in from a text file in the pattern:

Dividend

Divisor

And then passed to the divider component.

4 bit comparator input:



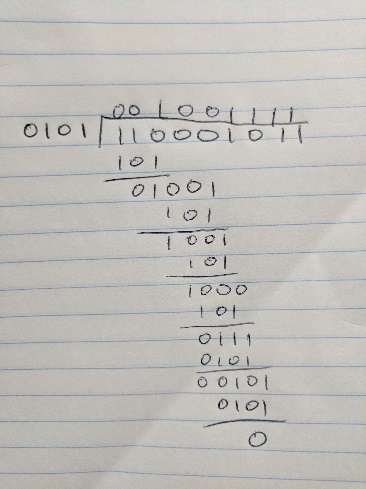
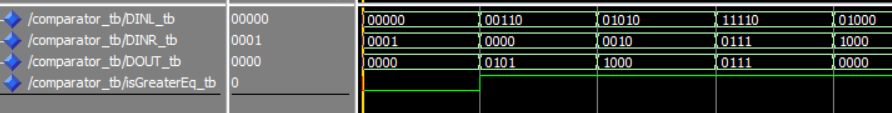


Figure 1 Binary Long Division by Hand

Waveform output:

16 bit divider simulation input: Output:

12

12 / 4 = 3 -- 0

16 / 2 = 8 -- 0

1000 / 0 = 1000 -- 0 OVERFLOW

12 / 7 = 1 -- 5

25000 / 250 = 100 -- 0

12 / 9 = 1 -- 3

3200 / 52 = 61 -- 28

58346 / 127 = 459 -- 53

12345 / 123 = 100 -- 45

9876 / 102 = 96 -- 84

4

16

2

1000

0

12

7

25000

250

12

9

3200

52

58346

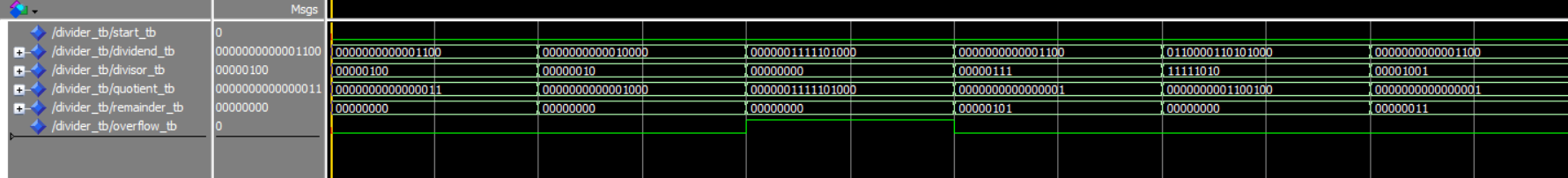
127

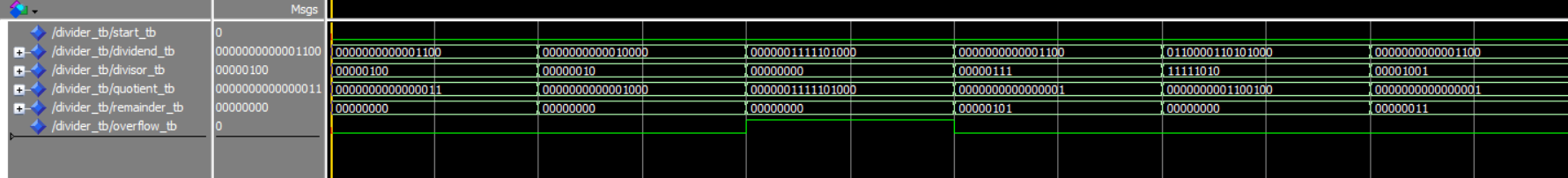
12345

123

9876

102

Waveform output:



32 bit divider simulation input: Output:

1000000

1000000 / 2500 = 400 -- 0

4598093 / 12300 = 373 -- 10193

1000009 / 4598 = 217 -- 2243

3200000 / 24501 = 130 -- 14870

3490000 / 9 = 387777 -- 7

123456 / 9876 = 12 -- 4944

4543245 / 6475 = 701 -- 4270

63665742 / 567 = 112285 -- 147

27538530 / 9852 = 2795 -- 2190

6575097 / 45 = 146113 -- 12

2500

4598093

12300

1000009

4598

3200000

24501

3490000

9

123456

9876

4543245

6475

63665742

567

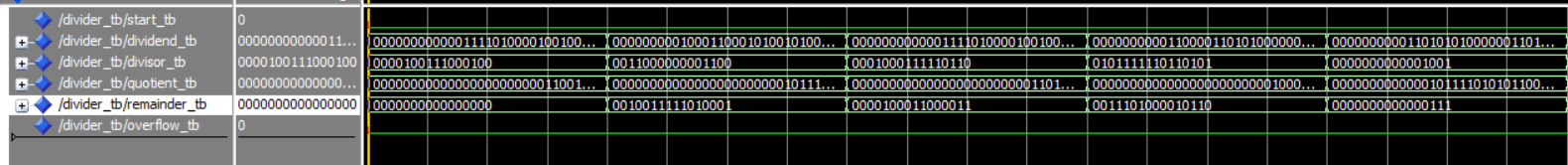
27538530

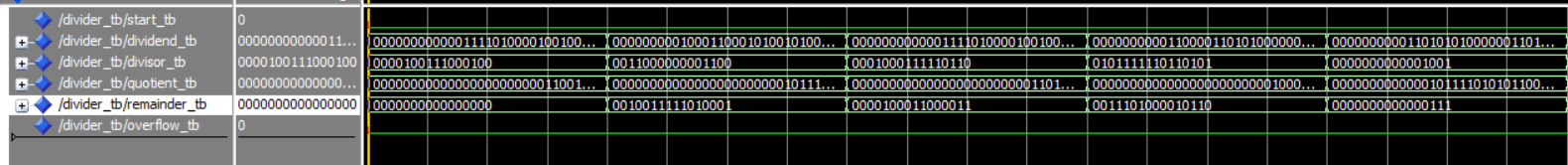
9852

6575097

45

Waveform output:





**Design**

The divider is built using a series of cascading comparators. The comparator is implemented as laid out in the lab pdf. For each comparator, the DINR input is always the divisor. An array of std\_logic\_vectors was made to store the output signals of each comparator, so that it could be fed in as an input to the next comparator. A generate statement was used to generate 32 comparators. The first comparator uses the most significant bit of the divisor, as the least significant bit of the DINL input. The rest of the DINL input is then set to zeros. The next 30 comparators take in the result of the comparator operation concatenated with the next bit in the dividend. This is easily implemented with DINL having one more bit than DOUT and DINR. This allows for the shifting and subtracting required for division. At each comparator, the isGreaterEq bit is connected to the corresponding output bit in the quotient. The final comparator outputs the result of its subtraction to the remainder rather than the intermediate output signal array. Overflow and the start button are handled with behavioral VHDL. Two signals were created called intDividend and intDivisor that are only updated when the start signal is high. These were the signals passed to the comparators, and they allow for the inputs to be changed without the output changing until the start button is pressed. Overflow was handled by checking if the intDivisor signal is zero, and then raising the overflow flag.