Graehme Blair

Drew Hasse

EECS 355

Lab 3: Sequential Divider

Simulation Results

Two simulations were run. Both the 16 and 32 bit test cases from Lab 2 were used to test the sequential divider.

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

Dividend

Divisor

And then passed to the divider component.

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

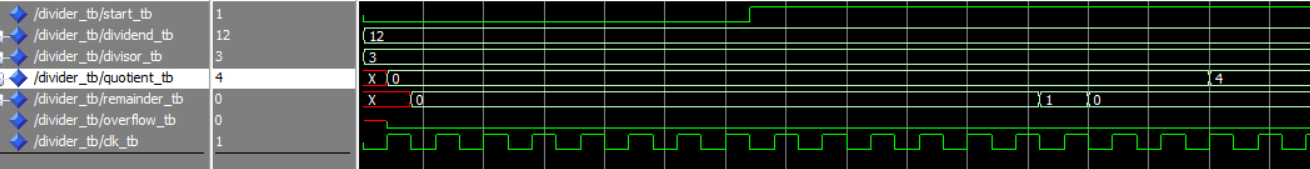
127

12345

123

9876

102

Waveform output (16 bit):

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

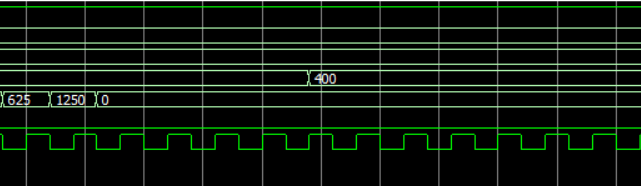
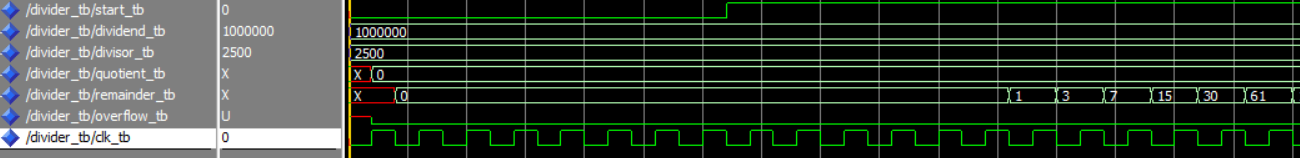
27538530

9852

6575097

45

Waveform output (32 bit):



Design:

The divider is implemented using one comparator. When the start button is pressed, the values dividend and divisor are passed to the comparator, the quotient is reset to zero, and the calculation begins. At each rising edge of the clock, remainder from the previous step is fed into the top bits of DINL and the next bit of the dividend is fed into the least significant bit of DINL. DINR is always set to the divisor. At the falling edge of the clock, the remainder is updated to DOUT from the comparator and the quotient bit is set in a variable holding the quotient during the calculation. A counter keeps track of the number of cycles, and after DIVIDEND\_WIDTH cycles, the quotient output is displayed.