3ba5 Computer Engineering

Michael Manzke,

Tuesday, 6th December 2005

Project 1b
VHDL Floating-point Pipeline Design

Design a VHDL floating-point pipeline which will compute Z = X + Y where X, Y and Z are 32-bite IEEE single precision non-zero floating-point numbers stored in the following format:

$$(-1)^{s} \times 2^{e} \times 1.m$$

The control of the pipeline is such that whenever an externally supplied signal ADD is asserted your circuit should add inputs X and Y to yield their sum Z, outputting one result every clock period once the pipeline has filled. In order to keep the project to a reasonable size we will constrain the input $X=(-1)^{xs} \times 2^{xe} \times 1.xm$ and $Y=(-1)^{ys} \times 2^{ye} \times 1.ym$ such that:

$$X,Y,Z \neq 0$$

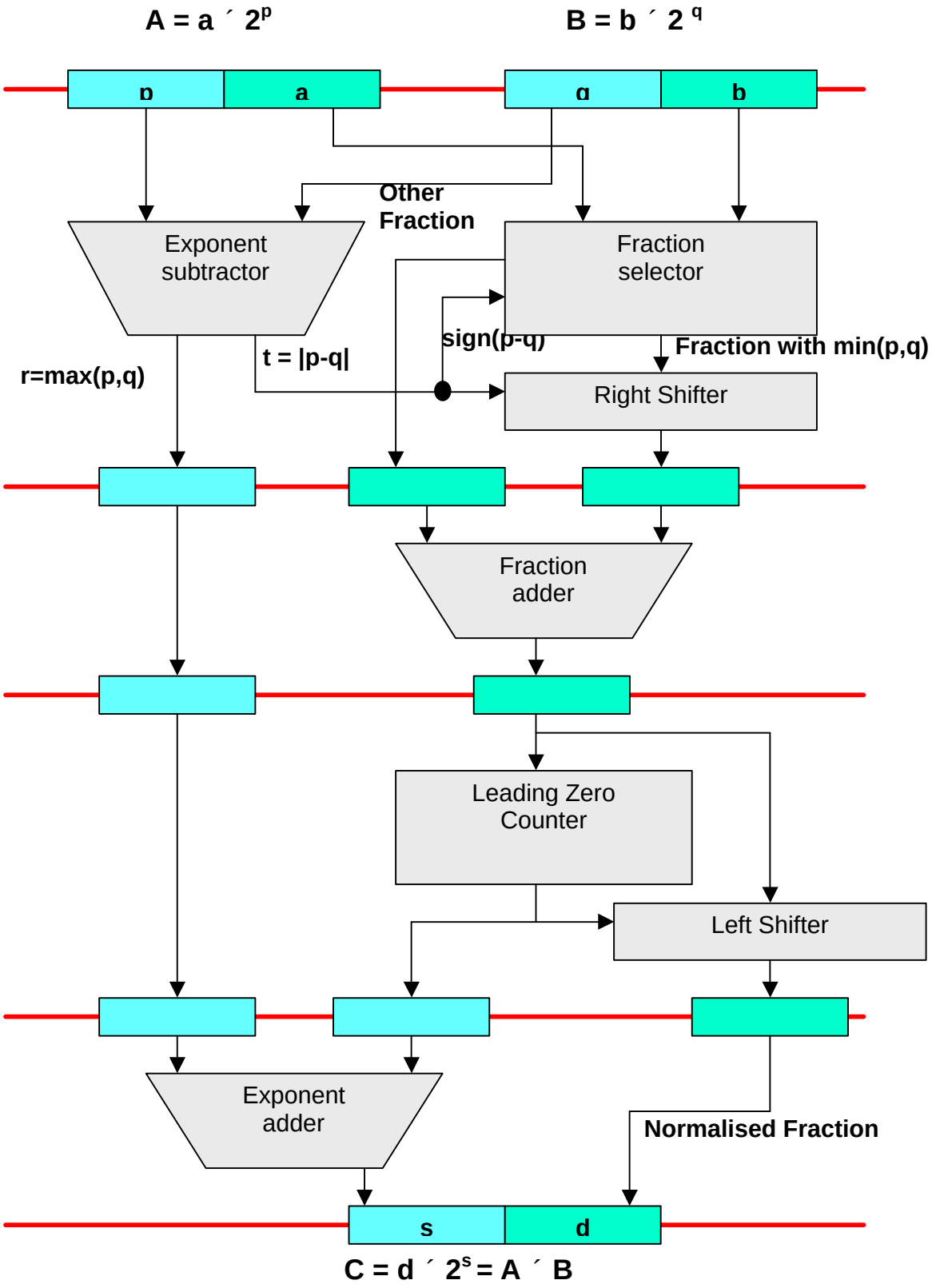
And

If $xs \neq ys$ then $2^{xe} \times 1.xm > 2^{ye} \times 1.ym$

Thus on account of a) there is no need to either detect or be able represent zero, and b) means that whenever a subtraction is required there is never a need to negate the new mantissa.

Simulate you VHDL design with ModelSim. Your project extends the first part of the project (1a) with a second source memory to provide the operands X and Y, and store your result Z in a in the Black Hole, which you implemented as part (1a) of this project. You may find the algorithm given overleaf showing the details of floating point addition/subtraction a useful guide, but of course you are free to develop your own design.

You should be able to show your pipeline adding n pairs of numbers and your test data should be carefully designed before you commence your implementation, starting with the simple's cases and progressing to the more complex cases so that each successive data test a specific part of your pipeline. Make sure that you manually compute the correct stage results for these test data and have them available for debugging. For each value of n=1, 4 and 16 compute the speedup of your pipeline. Try and establish what is the shortest clock period at which your pipeline will function properly and observe what happens when you shorten the clock period beyond this? You will be requested to demonstrate your simulation following submission of your project.


A = a ´ 2^p
B = b ´ 2^q
p
a
q
b
Other Fraction
Exponent subtractor
Fraction selector
t = |p-q|
sign(p-q)
Fraction with min(p,q)
r=max(p,q)
Right Shifter
Fraction adder
Leading Zero Counter
Left Shifter
Exponent adder
Normalised Fraction
s
d
C = d ´ 2^s = A ´ B

Project Deadline and Demo

Please submit a paper copy of your VHDL-code and test-bench-waves and an electronic copy of your project including all simulation results to the office in the Computer Science Department before 4:00pm on Monday 16th January 2006. You will have to demonstrate your project together with the first part on Wednesday, 18th January 2006 at 12:00-13:00 in LG35/36.

Late submissions will receive no credit.