Laboratory work 6 for TSTE18 Digital Arithmetic Approximation of Elementary Functions

Oscar Gustafsson

October 12, 2015

The purpose of this laboratory work is to realize different arithmetic operations with the help of a computer. The recommended programming languages to use are **Matlab**, **VHDL**, and **Verilog**. However, if you feel more comfortable with using another language feel free to do so, but make sure you have confirmed with the course responsibles that they can understand that programming language and have a fair chance of running your instances. **It is OK to use different programming languages for different parts/labs.**

It is required that you use a digit-based representation format for all your data, i.e., arrays, rather than integer types. Feel free to write conversion functions for easy use, but the main realization should use a digit-based representation in the appropriate radix.

The laboratories are **nominally individual**. However, as there are a few more students than can fit in a lab, it is OK to use the smallest number of pairs possible per lab. Note that each student is only allowed to work in pairs a minimum number of times. Accordingly, even though you happen to be working in a pair, you must make sure that you understand everything you submit. Also, note that the pairings will be decided in the lab, based on the number of students attending. Hence, there is no point in planning the pairs beforehand as some students may choose to do the labs at home. **All source codes will be cross-correlated, so please write your own and do not get "inspired"**.

On each sheet of paper write name, personal id number, and student-id, as well as the consecutive number assigned to you.

The reporting should consist of (printed and emailed to `oscarg@isy.liu.se`):

- Source code
- Example run, showing the usage of the realization
- Some non-trivial examples showing the correctness of the realization
- Where applicable, make sure that all relevant intermediate results are also shown in the examples
- Where applicable, drawings (hand-written is OK and no need to email those)

For Matlab, an .m-file with the runs and the output log would be fine. For VHDL/Verilog a .do-file setting up the windows etc and providing stimuli or a testbench may be appropriate.

> It is OK to have separate functions for the different cases.

1 CORDIC

Realize a function that performs a rotation of a complex number with a specified angle using a specified number of iterations (it is OK to maximize the number of iterations to, say, 20). There is no need to compensate for the gain error.

Plot the progress of an arbitrary rotation. In Matlab you can simply supply a vector of complex numbers, X, and do: `plot(X)`

2 Piecewise polynomial approximation

Determine a piecewise linear approximation of the function $\log_2\left(\cos\left(\frac{\pi}{2}X\right)\right)$ optimized in the minimax sense, i.e., minimize the largest error. Assume a ten bit binary input in the range $0 \leq X < 1$ and use four uniform segments. Use a fixed-point approximation of the coefficients with a reasonable number of bits. **Plot** the approximation and the correct curve in one plot and the approximation error in one plot.