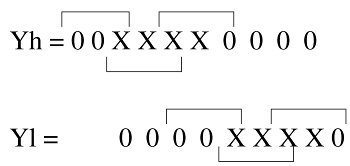
EE486: Advanced Computer Arithmetic
Homework #5 Solutions (35 pts)

(10pts) Problem 1 - Hybrid Modified Booth 2

One simple solution is to use:

X* (Yh – Yl)



The Yh table is the standard modified booth 2 table

The Yl table is shown below:

Yl i+1	Yl i	Yl i–1	Action
0	0	0	–0
0	0	1	–X
0	1	0	–X
0	1	1	–2X
1	0	0	+2X
1	0	1	+X
1	1	0	+X
1	1	1	+0

Six partial products must be summed together.
Three from the Yh and three from the Yl terms.

Figure 1: X*(Yh-Yl) Modified Booth 2 encoding

But a better method might try to handle the boundary region between Y_h and Y_l in one recoder. As a few of you tried this, you noticed that you will need to generate the "difficult" multiple $3X$. A solution that avoids even this hard multiple was presented in the original paper that needed this function of $X(Y_h - Y_l)$. If you are interested take a look at:

`http://arith.stanford.edu/papers.html`

and scroll down to: *Fast Division Algorithm with a Small Lookup Table*, Patrick Hung, Hossam Fahmy, Oskar Mencer, Michael J. Flynn, Asilomar Conference on Signals, Systems, and Computers, California, Nov. 1999.

Because it was used in a division algorithm, we gave you this problem in the homework on division and not on multiplication. The way the division is done in that paper is among the slightly more advanced ways about which you will hear in the coming few lectures.

(5pts) Problem 5.3 - Computation Using Newton-Raphson

$X_{i+1} = X_i(2 - bX_i)$

(2pts) Compute $1/b$ where $b = 0.9$

$$\begin{array}{ll} X_0 = 1 & E_0 = 0.11111111 \\ X_1 = 1.1 & E_1 = 0.01111111 \\ X_2 = 1.111 & E_1 = 0.00011111 \\ X_3 = 1.1111111 & E_1 = 0.00000001 \end{array}$$

(2pts) Compute $1/b$ where $b = 0.6$

$$\begin{array}{ll} X_0 = 1 & E_0 = 0.6666666 \\ X_1 = 1.4 & E_1 = 0.2666666 \\ X_2 = 1.624 & E_2 = 0.0426666 \\ X_3 = 1.6655744 & E_3 = 0.0010923 \\ X_4 = 1.666666 & E_4 = 0.0000006 \end{array}$$

(1pts) Compute $1/b$ where $b = 0.52$

$$\begin{array}{ll} X_0 = 1 & E_0 = 0.923076923 \\ X_1 = 1.48 & E_1 = 0.443076923077 \\ X_2 = 1.820992 & E_2 = 0.102084923077 \\ X_3 = 1.917657831 & E_3 = 0.005419092077 \\ X_4 = 1.923061653 & E_4 = 0.000015270076923 \end{array}$$

(10pts) Problem 5.5 - Alternative Newton-Raphson Reciprocal

(3pts) a) Find the iteration.

$$\begin{aligned} f(x) &= 1 + \frac{1}{b(x-1)} \\ f'(x) &= \frac{-1}{b(x-1)^2} \\ X_{i+1} &= bX_i^2 + 2(1-b)X_i + b - 1 \end{aligned}$$

(3pts) b) Compute the error term.

Let E_i be the error in iteration i so that $x_i = 1 - 1/b + E_i$, then

$$\begin{aligned} x_{i+1} &= b((1-1/b)^2 + 2(1-1/b)E_i + E_i^2) + 2(1-b)((1-1/b) + E_i) + b - 1 \\ x_{i+1} &= \frac{(b-1)^2}{b} + 2(1-b)(b-1)/b + b - 1 + (2b-2)E_i + bE_i^2 + (2-2b)E_i \\ x_{i+1} &= 1 - 1/b + bE_i^2 \\ E_{i+1} &= bE_i^2 \end{aligned}$$

(2pts) c) Compute 1/b with b=0.9 and b=0.6

For b=0.9	$X_0 = 0$	$(1 - X_0) = 1$	$E_0 = 0.11111111$
	$X_1 = -0.1$	$(1 - X_0) = 1.1$	$E_1 = 0.01111111$
	$X_2 = -0.111$	$(1 - X_0) = 1.111$	$E_1 = 0.00011111$
	$X_3 = -0.1111111$	$(1 - X_0) = 1.1111111$	$E_1 = 0.00000001$

For b=0.6	$X_0 = 0$	$(1 - X_0) = 1$	$E_0 = 0.6666666$
	$X_1 = -0.4$	$(1 - X_0) = 1.4$	$E_1 = 0.2666666$
	$X_2 = -0.624$	$(1 - X_0) = 1.624$	$E_2 = 0.0426666$
	$X_3 = -0.6655744$	$(1 - X_0) = 1.6655744$	$E_3 = 0.0010923$
	$X_4 = -0.6666666$	$(1 - X_0) = 1.666666$	$E_4 = 0.0000006$

(2pts) d) Compair this iteration to the standard Newton-Raphson Iteration

The error for both algorithms converge quadratically, $E_{i+1} = bE_i^2$.

If you start with the same initial point $X_{0_a} = 1$ for the standard reciprocal iteration and $X_{0_b} = (X_{0_a})^{comp} = 1 - X_{0_a} = 0$, then both produce the same result for each iteration.

The Standard iteration requires two multiplications and one add per iteration while the new algorithm requires one square, two multiply, and two adds per iteration. In this new algorithm, the bX_i^2 multiplication and the $2(1-b)X_i$ multiplication can be done in parallel if two multipliers exist. The results can be kept in a carry save form and added to the remaining $(b-1)$ term using only one CPA at the end. The delay is then approximately that of the square operation, one multiply, three CSAs and then a CPA. Depending on the implementation, a squaring unit may be faster than a regular multiplication and the new algorithm may then be faster than the standard iteration but requiring much more hardware. Hence, in general, the original algorithm seems to be better.

(10pts) Problem 5.7 - Newton-Raphson Cube Root

(4pts) a) Find the iteration.

$X_{i+1} = (2/3)X_i + (a/3)X_i^{-2}$

(2pts) b) Find the first two appriximations to $0.58^{1/3}$.

$X_0 = 1$, $X_1 = 0.86$, $X_2 = 0.8347359$

(4pts) c) Show the convergence (error term) for this iteration.

$$\begin{aligned}
X_i &= a^{1/3} + E_i \\
X_{i+1} &= (2/3)X_i + (a/3)X_i^{-2} \\
a^{1/3} + E_{i+1} &= (2/3)(a^{1/3} + E_i) + (a/3)(a^{1/3} + E_i)^{-2} \\
E_{i+1} &= \frac{2E_i}{3} - \frac{a^{1/3}}{3} + \frac{a}{3(a^{1/3} + E_i)^2}
\end{aligned}$$