EE 486 lecture 11: Divide and Square Root

M. J. Flynn

Fast algorithms

- Table lookup (large tables) using bipartite table(s) (lecture 15)
- Multiplicative (all use a small table to start)
  - Binomial series
  - Newton Raphson
  - Higher order series (lecture 15)

Multiplicative divide

- All approaches use (a) x (1/b), and find the reciprocal and then multiply by the numerator. Sometimes this multiply can be done earlier during the formation of the reciprocal.
- Two general approaches to finding 1/b
  - Series : $1/b = 1/(1+x) = 1-x+x^2-x^3$
  - NR: for $f(x)=0$ find x so that x is 1/b.

Binomial series

- $Q = a/b = a\ (1/b) = a\ (1/(1+x))$, this is expressible as as series $1/(1+x) = 1-x+x^2-x^3+x^4 \cdots =$ $(1-x)(1+x^2)(1+x^4)(1+x^8)\ldots.$
- $1-x =1-(b-1) =2-b$ and $1+x = b$; $C(1-x^2) =1+ x^2$
- Continue, forming $(1+x^4)(1+x^8)(1+x^{16})$..
- Since $x=b-1$; $0.5<=b < 1.0$; $-0.5<= x < 0$
- So a term like x^8 has 8 leading zeros and can affect the reciprocal in only the 9th place.
- Series is quadratically convergent (doubles precision each iteration.)

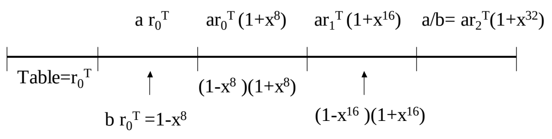
Starter tables

- Suppose we look up the first 8b of 1/b
- Same as $1/b = (1-x)(1+x^2)(1+x^4) +\varepsilon_0$: where ε_0 is $O(2^{-9})$. Since $b =1+x$
- $(1+x)\ ((1-x)(1+x^2)(1+x^4) +\varepsilon_0)= 1- x^8 +b\varepsilon_0$
- Now $(1- x^8 +b\varepsilon_0)(1+ x^8 -b\varepsilon_0)= 1- x^{16} +\varepsilon_1$
- So $\varepsilon_1 = 2b(b-1)^8\varepsilon_0 – b\varepsilon_0^2$
- And $\varepsilon_1 < 2^{-17}$

Starter tables

- Since b = 0.1xxxx, we use xxxx as table index, so if our 1/b is to be accurate to n bits
- $\varepsilon_0 >| 1/b – (1/(b-2^{-n})|$
- $\varepsilon_0 >| (b-2^{-n} -b)/(b^2-b2^{-n})|$
- So if we want ε_0 to be less than 2^{-9} ; then n is $b^2\,\varepsilon_0 < 2^{-n}$
- n =11, so table needs 10 bits, can be optimized by recognizing b = 0.5

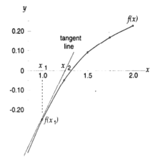
Timing



The initial reciprocal estimate is r_0^T

Newton-Raphson

- Find the root of $y=f(x)$
- $f'(x_0)=\Delta y/\Delta x$
- $f'=(0- f(x_0))/ (x_1- x_0)$
- $x_1= x_0 - f(x_0)/f'(x_0)$
- Now let $f(x)=(1/b) - x$
- $f'(x) = - (1/x^2)$
- So $x_{i+1}= x_i + x_i - bx_i^2$
- $x_{i+1}= x_i (2 - bx_i)$



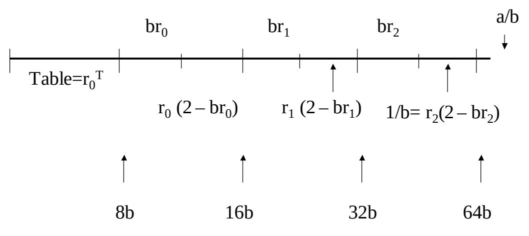
Newton Raphson error term

- $x_{i+1}= x_i (2 - bx_i)$; let $x_i = (1/b - \varepsilon_0)$ then
- $x_{i+1}= (1/b - \varepsilon_0) (2 – b(1/b - \varepsilon_0))$ so
- $= (1/b - \varepsilon_0) (1+ b\,\varepsilon_0)$
- $= (1/b - b\varepsilon_0^2)$
- So the error is negative and decreases quadratically.

Newton Raphson example:1/0.75

	iteration	q	error
0	$x_{i+1}= x_i (2 - bx_i)$	1.0	0.333334
1	1(2- 0.75)	1.25	0.083334
2	1.25(2- (1.25 x 0.75))	1.328125	0.005208
3	1.328125(2- (1.328125 x 0.75)	1.333313	0.000021

NR Timing



Timing considerations

- Typically, starter tables are about 8b; so we have 8, 16, 32, 64; 13b would be ideal for IEEE: 13, 26, 52, but 13b (8k or so) is difficult..
- The NR takes 2 multiplies/iteration; the binomial takes 2, but they can be overlapped using 2 multipliers: $(1+x^4)(1-x^4)$ can be done at the same time as $(1-x)(1+x^2)\,(1+x^4)$. The NR is preferred for vector divide, the binomial for iterative dividers.

Equivalence of binomial and NR

- For the NR, $x_{i+1} = x_i (2 - bx_i)$, let $x_o = 1$,
- So $1/b = (2-b)(2 – b(2-b)) = (2-b)(2-2b+b^2)$
- $= (2-b)(1+(1-b)^2)$
- Now the binomial series $b= 1+x$; $x= 1- b$
- So $1/b= (1-x)\ (1+x^2)(..) = (2-b)(1+(1-b)^2)$
- Both have the same term by term expansion
- Still they're not quite the same.

Remainder considerations

- Either approach gives the quotient and not the remainder. Since the iterations are the same the quotients are the same. Both fail to give the IEEE quotient and fail to protect "integers". (e.g. 1/0.8 = 1.2499999..).
- The remainder and quotient correction can be determined by 1- (b x q) = remainder; where the b x q product is 2n bits.

Remainder considerations (continued)

- The remainder and quotient correction can be determined by 1- (b x q) = remainder. The remainder must be positive in s+m.
- So if the leading n bits of the product are 1's (the b q product is 2n bits) we have a negative remainder-- then correct the q.

Hardware optimizations

- $(1+x)\ ((1-x)(1+x^2)(1+x^4))= 1- x^8$
- This is an 8 x 8 multiply and so is $(1-x^8)$

$(1+x^8)= 1- x^{16}$ so small multipliers or multiplies in a single multiplier are possible.

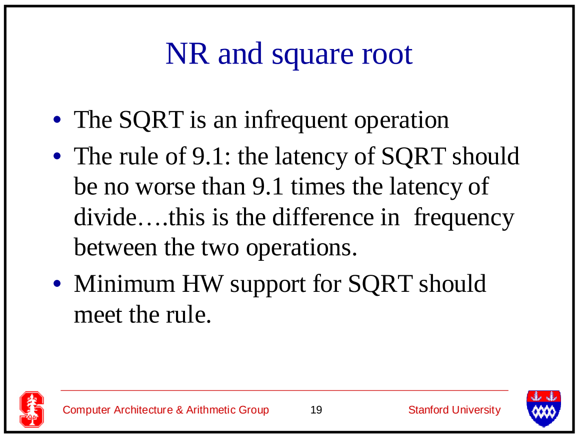
It is also possible to constrain the multiplier hardware so that the larger products take multiple passes.

NR and square root

- The simplest approach is to let $f(x) = b-x^2$
- Then $f'(x)= -2x$ and the iteration is
- $x_{i+1}= x_i/2+b/(2x_i)$; but this involves a divide which may be time consuming.
- An alternative is to find the root of the reciprocal of the square root. With that we can find the true square root by a multiply.
- Often the root reciprocal is the object, anyway.

NR and reciprocal of the SQRT

- Now let $f(x) = b – 1/x^2$ then $f'(x)= 2/x^3$
- So the iteration is $x_{i+1}= (x_i/2)(3 - b\ x_i^2)$;
- Also converges quadratically; needs 3 multiplies (compared to an add and a divide). Since it converges to 1/SQRT(b) a final multiply (by b) may be needed to form the SQRT(b).



NR and square root

- The SQRT is an infrequent operation
- The rule of 9.1: the latency of SQRT should be no worse than 9.1 times the latency of divide….this is the difference in frequency between the two operations.
- Minimum HW support for SQRT should meet the rule.