



UNIVERSITY OF MASSACHUSETTS
Dept. of Electrical & Computer Engineering

Digital Computer Arithmetic
ECE 666

Part 8
Division through Multiplication

Israel Koren
Spring 2008

ECE666/Koren Part. 8.1

Copyright 2008 Koren

Division by Convergence

- ◆ Number of steps proportional to $\log_2 n$
- ◆ Basic operation - multiply ; fast parallel multiplier necessary
- ◆ $Q=N / D$ - same quotient if numerator and denominator multiplied by R_0, R_1, \dots, R_{m-1}
- ◆ R_i 's selected so that denominator converges to 1
 \Rightarrow numerator converges to Q :

$$Q = \frac{N}{D} = \frac{N \cdot R_0 R_1 \cdots R_{m-1}}{D \cdot R_0 R_1 \cdots R_{m-1}} \rightarrow \frac{Q}{1}$$

- ◆ Only quotient calculated - separate computation needed for remainder
- ◆ Scheme more suitable for floating-point division

ECE666/Koren Part. 8.2

Copyright 2008 Koren

Selection of Factors

- ◆ Factors selected so that denominator converges to 1
- ◆ D - normalized binary fraction $0.1xxxx$ (x is 0 or 1)
 - * $1/2 \leq D < 1 \Rightarrow D = 1-y$ with $y \leq 1/2$
- ◆ **Step 1: Select $R_0 = 1+y$**
 - * New denominator: $D_1 = D \cdot R_0 = (1-y) \cdot (1+y) = 1-y^2$
 - * $y^2 \leq 1/4 \Rightarrow D_1 \geq 3/4$ - closer to 1 than D
 - * $D_1 = 0.11xxxx$
- ◆ **Step 2: Select $R_1 = 1+y^2$**
 - * New denominator: $D_2 = D_1 \cdot R_1 = (1-y^2) \cdot (1+y^2) = 1-y^4 \geq 15/16$
 - * $D_2 = 0.1111xxxx$ - closer to 1 than D_1
- ◆ **Step $i+1$: $D_i = 1-y_i$ where $y_i = y^{2^i}$**
 - * At least 2^i leading 1's
 - * Next multiplying factor $R_i = 1+y_i$; $D_{i+1} = D_i \cdot R_i$ has at least 2^{i+1} leading 1's - closer to 1

ECE666/Koren Part. 8.3

Copyright 2008 Koren

Formal Proof of Convergence

- ◆ $(1-y) \cdot [(1+y)(1+y^2)(1+y^4) \dots] = (1+y) \cdot [(1-y)(1+y^2)(1+y^4) \dots]$
- ◆ Term within brackets on right - series expansion of $1/(1+y)$ for $0 \leq y \leq \frac{1}{2}$
- ◆ $\lim_{i \rightarrow \infty} D_i = (1+y) \cdot 1/(1+y) = 1$
- ◆ Multiplying by R_i repeated until D_i converges to 1 - more precisely, to $0.11\dots 1$ ($=1-ulp$)
- ◆ Number of leading 1's in D_i doubled at each step
- ◆ Number of iterations is $m = \lceil \log_2 n \rceil$
 - * Quadratic convergence
- ◆ Multiplying factor R_i obtained from D_i
- ◆ $R_i = 2 - D_i$ - two's complement of fraction D_i
- ◆ Each step consists of 2 multiplications: $D_{i+1} = D_i \cdot R_i$ & $N_{i+1} = N_i \cdot R_i$ and two's complement $R_{i+1} = 2 - D_{i+1}$

ECE666/Koren Part. 8.4

Copyright 2008 Koren

Example: 15-bit Numbers

- ◆ $N=0.011,010,000,000,000 = 0.40625_{10}$
- ◆ $D=0.110,000,000,000,000 = 0.75_{10}$
- ◆ $R_0=2-D=1.010,000,000,000,000$
- ◆ $N_1=N \cdot R_0=0.100,000,100,000,000$
- ◆ $D_1=D \cdot R_0=0.111,100,000,000,000$
- ◆ $R_1=2-D_1=1.000,100,000,000,000$
- ◆ $N_2=N_1 \cdot R_1=0.100,010,100,010,000$
- ◆ $D_2=D_1 \cdot R_1=0.111,111,110,000,000$
- ◆ Number of leading 1's in D_2 doubled from 4 to 8
- ◆ $R_2=2-D_2=1.000,000,010,000,000$
- ◆ $N_3=N_2 \cdot R_2=0.100,010,101,010,101$
- ◆ $D_3=D_2 \cdot R_2=0.111,111,111,111,111$
- ◆ Convergence ($D_3=1-\text{ulp}$) in 3 steps
- ◆ $Q=N_3=0.54165_{10}$ - exact result is infinite fraction 0.54166_{10}

ECE666/Koren Part. 8.5

Copyright 2008 Koren

Speed-Up Techniques

- ◆ Total number of steps: order of $\log_2 n$
 - * Algorithms based on add/subtract: linear in n
- ◆ Each step - 2 multiplications
- ◆ Need to further reduce number of steps
- ◆ Speed up first few steps - slow convergence
 - * After step 1, only two leading 1's guaranteed
 - * After step 2 - only four 1's
- ◆ Instead of $R_0 = 1+y$, use a look-up table - multiplier ensuring D_1 with k leading 1's ($k \geq 3$)
- ◆ Next denominator with $2k$ leading 1's, and so on
- ◆ Table size (in ROM) increases exponentially with k
- ◆ k determined so that table size is reasonable

ECE666/Koren Part. 8.6

Copyright 2008 Koren

Example: IBM 360/91 Floating-Point Division

S	7 bits - biased exponent	56 bits - unsigned fractional significand
---	--------------------------	---

- ◆ If $R_0 = 1+y$, $\lceil \log_2 56 \rceil = 6$ steps needed
 - * Requiring 12 multiplications of 56 bits each
 - * Only 11 - no need to calculate D_5 in last step
- ◆ Look-up table for R_0 so that D_1 has at least $k=7$ leading 1's, yielding: $1 \rightarrow 7 \rightarrow 14 \rightarrow 28 \rightarrow 56$
 - * Only 4 steps requiring 7 multiplications
- ◆ 7 bits of D needed and 10 bits of R_1 stored at each location: ROM of size 128×10
- ◆ Row corresponding to $D=1-y$ - 10-bit approximation of $(1+y)(1+y^2)(1+y^4)$
- ◆ At full precision would get 8 leading 1's
- ◆ No error - multiplies both numerator and denominator
 - * Previous convergence scheme initiated at this point

ECE666/Koren Part. 8.7

Copyright 2008 Koren

Speed Up by Using Shorter Multipliers

- ◆ Use truncated multipliers for some products
- ◆ No error - numerator and denominator multiplied by same factor
- ◆ Not for last product - high accuracy is needed
- ◆ **Step $i+1$:** generate D_{i+1} with $a (\geq 2^{i+1})$ leading 1's by multiplying D_i ($a/2$ leading 1's) by R_i
- ◆ Instead of $R_i = 2 - D_i = 1 + y_i$ use truncated R_{iT} - two's complement of first a bits of D_i - truncated D_{iT}
- ◆ $R_{iT} = 2 - D_{iT}$; denote $R_{iT} = 1 + y_T \rightarrow$ error in truncated multiplier is $\alpha = y_T - y_i - 0 \leq \alpha < 2^{-a}$
- ◆ Multiplying truncated multiplier by untruncated denominator:

$$D_{i+1} = D_i \cdot R_{iT} = (1 - y_i) \cdot (1 + y_T) = 1 + y_T - y_i - y_i y_T$$

ECE666/Koren Part. 8.8

Copyright 2008 Koren

Shorter Multipliers - Resulting "Error"

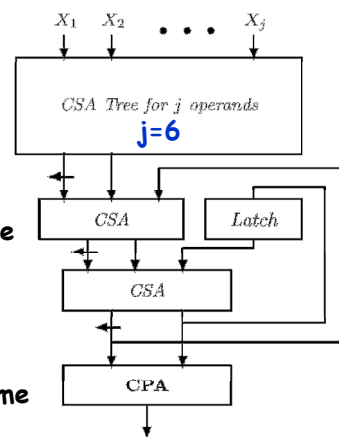
- ◆ Substituting $y_T = y_i + \alpha$:

$$D_{i+1} = 1 + \alpha - y_i(y_i + \alpha) = 1 - y_i^2 + \alpha(1 - y_i)$$

- ◆ "Error" in D_{i+1} is $\alpha(1 - y_i)$ -
 $0 \leq \alpha(1 - y_i) \leq \alpha < 2^{-a}$
 - * "Error" is always positive
- ◆ D_{i+1} still has a leading 1's - may converge toward 1 from below/above
 $(D_{i+1}=0.11\dots1xxxx/D_{i+1}=1.00\dots0xxxx)$
- ◆ In truncated multiplication factors - first half of bits identical - all 0's or all 1's
- ◆ If multiplier R_i recoded in SD - leading 0's or 1's will not generate nonzero partial products
 - * Execution time of multiplication further reduced

Example - Floating-Point Fast Multiplier in IBM 360/91

- ◆ Operands of 56 bits - uses algorithm in Table 6.5
- ◆ Generates partial products $0, \pm 2A, \pm 4A$
- ◆ 28 partial products require 26 carry-save adders
- ◆ Carry-save tree for 8 operands
 - * 6 new partial products added to 2 previous intermediate results
 - * Used 5 times
 - * Pipeline allows overlapping consecutive sets of 6 partial products
 - * Accumulating all 28 partial products takes 6 clock cycles
 - * Overlapping possible among sets of partial products corresponding to same multiply operation



Example - Overlapping Multiplications

- ◆ Overlapping 2 different multiply ops achievable if number of generated partial products ≤ 6
- ◆ Carry-save tree passed only once per multiply operation - no need to use feedback connections
- ◆ Limiting to 6 partial products (or less) can speed up execution of consecutive multiplications
- ◆ To get sequence of D_i with 7, 14, 28, 56 leading 1's (or 0's) - need multipliers R_i with 10, 14, 28, 56 bits
- ◆ First multiplier - out of ROM - generates only 5 partial products
- ◆ Other 3 multipliers contain 7, 14, and 28 leading 0's (or 1's) which can be skipped
 - * No need to generate partial products
 - * Just identify first and last bits of group of identical bits

ECE666/Koren Part. 8.11

Copyright 2008 Koren

Example - Further Multiplier Truncation

- ◆ Second multiplier (14 bits), generates only 5 partial products - feedback connections not used

$$0. \underbrace{11}_{1x} \quad 11 \quad 11 \quad \underbrace{1x}_{xx} \quad \underbrace{xx}_{xx} \quad \underbrace{xx}_{xx}$$

- ◆ 3rd multiplier (28 bits), generates 9 partial products, requiring use of feedback connection
- ◆ Can be avoided by additional truncation of multiplier
- ◆ Can add 9 bits to 14 leading identical bits (total of 23 bits) - still only 6 partial products:

$$0. \underbrace{11}_{1x} \quad 11 \quad 11 \quad 11 \quad 11 \quad 11 \quad 11 \quad \underbrace{1x}_{xx} \quad \underbrace{xx}_{xx} \quad \underbrace{xx}_{xx} \quad \underbrace{xx}_{xx}$$

- ◆ New denominator only guaranteed to have $14+9=23$ leading identical bits instead of 28
- * **Proof** - exercise

ECE666/Koren Part. 8.12

Copyright 2008 Koren

Example - Summary

- ◆ Next multiplier - only 23 leading identical bits - can add 9 extra bits without use of feedback
- ◆ Denominator has 23+9=32 leading identical bits
- ◆ Two's complement of it → multiplier with 32 leading identical bits - number of leading identical bits in denominator increases to 64 and convergence achieved
- ◆ Last multiply operation - feedback used - all available 56 bits used
- ◆ Sequence of 5 multiplication factors of length 10,14,23,32,56 bits, increasing number of multiply operations from 7 to 9
- ◆ All can be overlapped -total execution time of 18 clock cycles

ECE666/Koren Part. 8.13

Copyright 2008 Koren

Division by Reciprocation

- ◆ Reciprocal of divisor D multiplied by dividend
- ◆ Reciprocal calculated using Newton-Raphson iteration method - finding zero of function $f(x)$ (solution of $f(x)=0$)
- ◆ x_0 - first approximation ; x_i - i th step estimate for zero ; $f'(x)$ - derivative of $f(x)$

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

- ◆ $f(x)=\{1/x\}-D$ has a zero at $x=1/D$
- ◆ $f'(x)=-1/x^2$
- ◆ $x_{i+1} = x_i (2-D x_i)$
- ◆ x_{i+1} converges to reciprocal of D
- ◆ Convergence is quadratic

ECE666/Koren Part. 8.14

Copyright 2008 Koren

Proof of Convergence

- ◆ $\Delta_i = 1/D - x_i$ - error in i th step
- ◆ Simple algebraic manipulations - $\Delta_{i+1} = D \Delta_i^2$
- ◆ D normalized fraction ($\frac{1}{2} \leq D < 1$) $\Rightarrow \Delta_i \leq 1$ - error decreases quadratically
- ◆ $x_0 = 1$, $x_1 = 2 - D$, $x_2 = (2 - D) \cdot [2 - D(2 - D)] = (2 - D) \cdot [1 + (D - 1)^2]$

◆ Repeatedly substitution results in

$$\begin{aligned} x_i &= (2 - D)(1 + (D - 1)^2)(1 + (D - 1)^4) \cdots (1 + (D - 1)^{2^i}) \\ &= (1 - (D - 1))(1 + (D - 1)^2)(1 + (D - 1)^4) \cdots (1 + (D - 1)^{2^i}) \end{aligned}$$

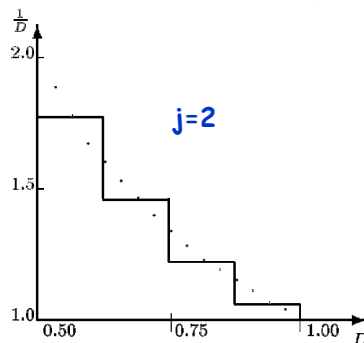
◆ $y = 1 - D$; $0 < y \leq \frac{1}{2}$

$$\lim_{i \rightarrow \infty} x_i = \frac{1}{1 + (D - 1)} = \frac{1}{D}$$

Reducing Number of Steps

- ◆ Using table for first step rather than $x_0 = 1$
- ◆ Table (ROM) accepts j most significant digits of D (except first = 1) - produces approximation to $1/D$
- ◆ Range $[0.5, 1)$ divided into 2^j intervals (of size $\Delta = 0.5 \cdot 2^{-j}$) - optimum value of x_0 for k th interval ($k = 1, 2, \dots, 2^j$) is reciprocal of middle point of interval

Middle point is $\frac{1}{2} + (k - \frac{1}{2})\Delta$



$$x_0(k) = \frac{2^{j+1}}{2^j + k - \frac{1}{2}}$$

Piecewise linear approximation
- more complicated but higher accuracy

Implementation in the 64-bit ZS-1

- ◆ Uses **IEEE** floating-point - significand $1 \leq d < 2$
- ◆ **15** most significant bits of divisor (excluding hidden bit), $1.d_1 d_2 \dots d_{15}$, address a **ROM** look-up table for initial approximation x_0
- ◆ Table size **32Kx16** bits - $0.5 \leq x_0 < 1$
- ◆ $x_0 = 0.1y_2 y_3 \dots y_{16}$
- ◆ Approximation is reciprocal of mid-point between $1.d_1 d_2 \dots d_{15}$ and its successor - $1.d_1 d_2 \dots d_{15} 1$
- ◆ Reciprocal rounded by adding 2^{-17} - result truncated to yield 16 bits: $0.1y_2 y_3 \dots y_{16}$
- ◆ Precision: $|x_0 - 1/d| < 1.5 \cdot 2^{-16}$
- ◆ Two iterations needed to achieve precision of 53 bits
 - * Four multiplications and two complement operations

ECE666/Koren Part. 8.17

Copyright 2008 Koren

Reducing Execution Time

- ◆ First iteration: $x_1 = x_0 (2 - d x_0)$
- ◆ 16 bits of x_0 multiplied by 32 most significant bits of d - result rounded to 32, instead of 48, bits
- ◆ One's complement - avoid carry-propagation
 - * Introduces an error of size 2^{-31}
- ◆ Multiply 16 bits of x_0 by 32 bits of multiplicand \rightarrow 32-bit product $\rightarrow x_1$ accurate to approx. 31 bits
- ◆ Second iteration: similar operations
- ◆ In $x_1 d$, only 64 bits generated and only one's complement calculated - $x_1 (2 - d x_1)$ performed producing approximated value of $1/d$
- ◆ Multiply by N and approximated Q' rounded (one of four IEEE rounding schemes)
 - * Final rounding does not guarantee an accurately rounded result for all values of d

ECE666/Koren Part. 8.18

Copyright 2008 Koren

Accuracy

- ◆ Most implementations of division-by-reciprocation - accuracy smaller than for add/subtract type division
- ◆ Corrective actions can be taken to guarantee correctly rounded least significant bit
- ◆ Additional computation slows down division
- ◆ When deciding on a division algorithm consider: precision, speed, and cost tradeoffs
 - * Final decision depends on available technology
- ◆ Precision required by IEEE standard achieved at a reasonable cost and speed in IBM RISC/6000
- ◆ Double-width datapath - all operations done in a fused multiply-add unit resulting a double-length estimate Q' of quotient

IBM RISC/6000

- ◆ Remainder $R=N-DQ'$ (fused multiply-add) - used to compute properly rounded result (fused multiply-add) in desired rounding mode
- ◆ $Q=Q'+R \times 1/D$ - can use fused multiply-add unit
- ◆ $1/D$ result of Newton-Raphson iterations
- ◆ Different solution: estimate error in Q' by calculating $N'=D Q'$
- ◆ Q' is sufficiently accurate (at least to $n+1$ bits; n is number of bits in significand), least significant bits of N' provide direction of error in Q'
- ◆ Based on this and desired rounding mode, Q' can be corrected by either adding or subtracting 1 at the $n+1$ bit position or by truncating it