A Powering Unit for an OpenGL Lighting Engine

David Harris

David_Harris@hmc.edu

Harvey Mudd College

Claremont, CA

35th Asilomar Conference on Signals, Systems, and Computers

November 2001

Introduction

OpenGL Transformation & Lighting Pipeline

- Specular lighting and spotlights require powering operation $P = A^B$
- $A \in [0, 1]$, $B \in [1, 128]$
- Results must be accurate to color depth (8-10 fractional bits)

Use identity $A^B = 2^{B \log_2 A}$

- Requires log table lookup, multiplication, and exponent table lookup
- Log tables are very large for 8-10 bit accuracy
- Accuracy requirements increase as A approaches 1
- Partition log table into subintervals with increasing accuracy

Synthesis results

Algorithm

Compute $P = A^B = 2^{B \log_2 A}$

- $A \in [0, 1]$, $B \in [1, 2^b]$ provided as IEEE single-precision FP numbers
- $P \in [0, 1]$ is faithfully rounded to p-bit fraction and expressed as FP number

$L = \log_2 A$

- Use n_1 bits of significand field of A to look up fractional part of logarithm L
- Exponent field of A becomes integer part of logarithm L

$X = B \bullet L$

- Use n_2 fractional bits of B and n_3 fractional bits of L

$P = 2^X$

- Use n_4 fractional bits of X to look up significand of P to n_5 = p
- Use integer part of X to determine exponent of P

Error Analysis

Finite numbers of fractional bits introduce errors at each step

- n_1 bits of A index log table error ε_1
- n_2 bits of B provided to multiplier error ε_2
- n_3 bits of L produced by log table error ε_3
- n_4 bits of X returned by multiplier error ε_4
- n_5 bits of B returned by exp table error ε_5

Instead of computing

$$P = A^B = 2^{B \log_2 A}$$

Actually compute

$$P = 2^{(B + \varepsilon_2)(\log_2(A + \varepsilon_1) + \varepsilon_3) + \varepsilon_4} + \varepsilon_5$$

Choose n's so error $\left|P - A^B\right| < 2^{-p}$

Table Sizes

Taylor series approximations can be used to find impact of *n*'s on error

$$P = 2^{(B+\varepsilon_2)(\log_2(A+\varepsilon_1)+\varepsilon_3)+\varepsilon_4} + \varepsilon_5$$

$$\left|P - A^B\right| \approx \left|A^B\left(\varepsilon_1\frac{B}{A} + \varepsilon_2 \log_2 A \ln 2 + \varepsilon_3 B \ln 2 + \varepsilon_4 \ln 2\right) + \varepsilon_5\right| < 2^{-p}$$

After some analysis, choose:

- $n_1 = p + b + 1$ bits of *A* to index logarithm table
- $n_2 = p + 3$ bits of *B* for multiplier
- $n_3 = p + b + 4$ bits of *L* for multiplier
- $n_4 = p + 2$ bits of *X* to index exponent table
- $n_5 = p$ *P* faithfully rounded to *p* bits

For $p = 10$, $b = 7$, this requires a log table of 256k entries

A Closer Look at Logarithm Table Errors

Log lookup must be very accurate for *A* just under 1

- $L = log_2 A$ will have many leading 0's that cancel when multiplied by large *B*
- $\left|P - A^B\right| \approx \left|A^B\left(\varepsilon_1 \frac{B}{A} + \varepsilon_2 log_2 A \ln 2 + \varepsilon_3 B \ln 2 + \varepsilon_4 \ln 2\right) + \varepsilon_5\right| < 2^{-p}$

Maximum weight on ε_1 term (for *b* = 7)

Range of A	maximum value of BA^{B-1}
[0, 0.5)	1.07
[0.5, 0.75)	1.71
$[1-2^{-2}, 1-2^{-3})$	3.15
$[1-2^{-3}, 1-2^{-4})$	6.08
$[1-2^{-4}, 1-2^{-5})$	12.0
$[1-2^{-5}, 1-2^{-6})$	23.7
$[1-2^{-6}, 1-2^{-7})$	47.3
$[1-2^{-7}, 1-2^{-8})$	77.9
$[1-2^{-8}, 1)$	128

Multiple Logarithm Tables

The maximum weight on the ε_1 term increases with A

- Thus more bits of A are required to index the table for larger values of A

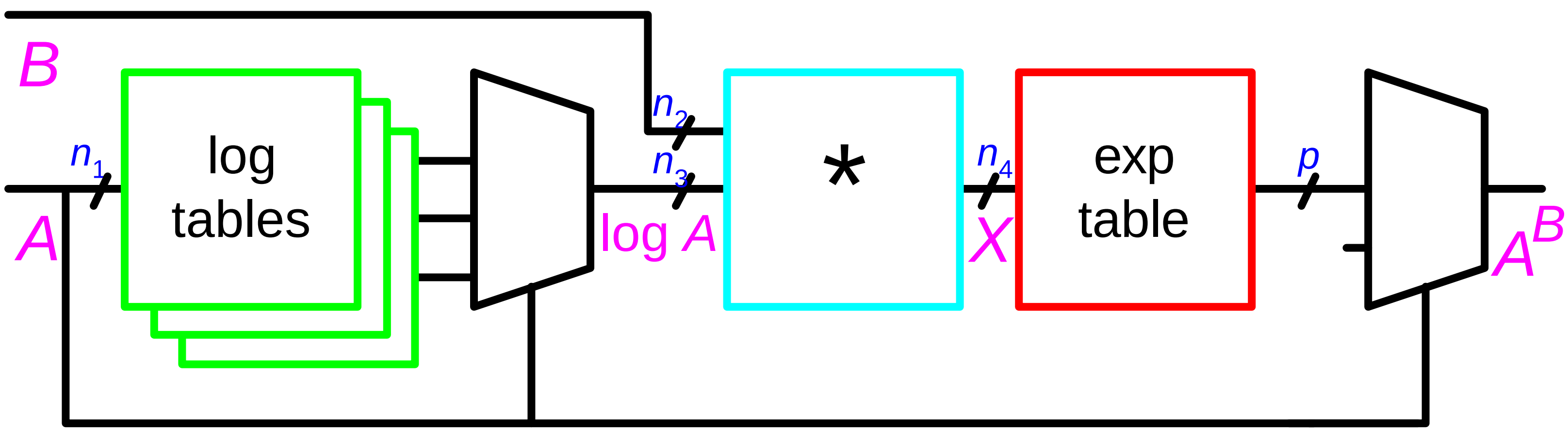
Partition log table into $b + 2$ subintervals covering progressively smaller ranges

- Table T_i covers A in range $[1-2^{-i}, 1-2^{-(i+1)})$ for $i = 0...b$
- Table T_{b+1} covers $[1-2^{-b}, 1)$
- Leading i bits of A are all 1's
- Index table with next $\tilde{n}_1 = p$ bits

Now we only need $b + 2$ logarithm tables of 2^p entries each

- For $p = 10$, $b = 7$, this requires 9k total entries

Architecture



Implementation

Design implemented in Verilog and compared against C reference model

- Parameterized by *p* and *b*
- Tested for $p = 8, 10$; $b = 7$ (OpenGL application)

Verification

- 6 million directed and random test vectors used to verify accuracy
- For $p = 8$, maximum error = $0.0029 < 2^{-p} = 0.0039$
- For $p = 10$, maximum error = $0.00076 < 2^{-p} = 0.00098$
- Faithful rounding confirmed for all test cases

Source Code

- Verilog and C models are on the web
- Harvey Mudd College Open Source Floating Point Project
- `www.hmc.edu/chips`

Synthesized to LSI G12-p 180 nm standard cell library

Synthesis Results

Latency:

- 7.87 ns for $p = 8$
- 9.62 ns for $p = 10$
- Could be partitioned into three stage pipeline

Area:

Powering Unit Gate Count

Module	$p = 8$	$p = 10$
Log Tables	14 867	72 734
Exp Tables	1 823	6 181
Multiplier	2 317	3 035
Random Logic	1 176	1 669
Total	20 183	83 619

- At an estimated 20-25K gates / mm^2, area is 1 to 4 mm^2

ROMs

Synthesized tables are very inefficient

Table Bit Counts

Table	Size	$p = 8$, $b = 7$	$p = 10$, $b = 7$
logarithm	$2^p[(b+2)(p+4)]$	28 416	132 096
exponent	$p2^{p+2}$	8 192	40 960

A ROM generator would greatly reduce table area

- Estimated unit size < 0.6 mm^2 for $p = 10$

Conclusion

Hardware implementation of Powering Unit

- Computes $P = A^B$ for $A \in [0, 1]$, $B \in [1, 128]$
- Optimized for OpenGL lighting calculations with 8-10 bits of accuracy
- Useful for other low-precision applications

Use identity $A^B = 2^{B\log_2 A}$

- Reduce size of logarithm tables by partitioning into subintervals

Verilog and C models used for verification and synthesis

- For 10-bit accuracy:
  - 9 1024-entry log tables and one 2048-entry exponent table
  - area = 4mm^2 synthesized or about 0.6 mm^2 with ROMs
  - latency: 9.62 ns

Source code available through HMC Open Source Floating Point Project