Architectures for Floating-Point Division

Hooman Nikmehr

B.Sc. University of Tehran

M.Eng.Sc. University of Tehran

A thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

in the

School of Electrical and Electronic Engineering

The University of Adelaide

Australia

Supervisors: **Dr. Cheng-Chew Lim** and **Dr. Braden Phillips**

August, 2005

Contents

List of Figures

List of Tables

Abstract

Almost all recent microprocessors and DSP chips perform addition, subtraction, multiplication and division in hardware. However, studying their performance reveals that division is not carried out as fast as the other three operations. One investigation shows that while floating-point division, with about 3% of the dynamic floating-point instruction count, seems to be a relatively unimportant instruction, it may cause about 40% degradation to the overall system performance.

Several mathematical algorithms have been developed over the past 50 years to perform division quickly, with high precision. However, only a few are suitable for implementation in VLSI. Among them, digit recurrence algorithms are the most widely accepted methods of performing floating-point division in the latest processors. A survey shows that out of 13 recent processors, 11 use SRT division[1] for performing floating-point division. Investigations show that SRT division gives the best trade-off between delay and area. Selecting SRT division for implementing floating-point division is a reasonable choice because, unlike the other class of division algorithms, i.e. functional, it produces a correctly rounded quotient conforming to the IEEE 754 standard.

There are techniques for improving the performance of SRT division. Of these, increasing the speed of quotient digit selection (QDS), making the best balance between the radix and the redundancy factor, representing the partial remainder in a redundant form, converting the quotient from redundant to conventional form the on-the-fly and overlapping the division recurrence components are the most important.

In this thesis a different method of implementing the QDS function is proposed. This approach, which is described mathematically and architecturally, is based on the new *comparison multiples* idea. Unlike the traditional implementation of the QDS function, which searches for the quotient digit in a lookup table, the proposed method calculates

[1]SRT division is a type of non-restoring digit recurrence division.

the quotient digit directly in sign and magnitude format. This approach almost halves the fan out of some critical path components, which therefore operate faster. Having received the truncated partial remainder, the QDS function compares it with truncated multiples of the divisor to find the range in which the partial remainder belongs. The results of the comparisons are converted to the magnitude of the quotient digit using simple logic called the coder. Concurrently, another circuit checks the truncated partial remainder to determine whether the quotient digit is negative. This circuit operates off the critical path since the comparison multiples based QDS function calculates the sign and magnitude of the quotient digit separately. Having applied these changes, a faster QDS function and consequently, a shorter critical path delay for the floating-point divider is obtained. Implementations of radix-4 and radix-16 floating-point dividers are investigated and optimised to further decrease the cycle time.

The idea of comparison multiples is extended to radix 10 to implement a decimal floating-point divider complying with the IEEE 754R standard. To achieve this goal, decimal signed-digit arithmetic along with implementations of carry-free addition and subtraction are proposed. The original comparison multiples based implementation of high-radix SRT division is modified to suit radix 10.

The binary and decimal implementations of comparison multiples based division are evaluated for delay. Using the method of logical effort, the radix-4, radix-16 and decimal floating-point dividers are found to be faster than corresponding circuits reported in the public literature.

Statement of Originality

I hereby declare that this work contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution and to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text.

I give consent to this copy of my thesis, when deposited in the University Library, being available for loan and photocopying.

Hooman Nikmehr
15 August 2005

Acknowledgments

I would like to thank my supervisors, Dr. Cheng-Chew Lim and Dr. Braden Philips, who have technically and mentally supported me during my PhD research. Dr. Lim taught me how to successfully pass academic milestones by hardwork and punctuality and Dr. Philips opened my eyes to different perspectives of the design for digital arithmetic.

I would also like to express my sincere thanks to Mr. Ron Seidel, who is one of the best friends I have ever had. I have benefited from his priceless advice to cope with the problems, fears and confusion I have had during my residency in Australia.

I must express my gratitude to my mother who took care of the bureaucracy related to my scholarship in Iran.

Finally, without the endless love, intense passion and inexpressible patience that my wife, Mehrnaz, has offered me, even dreaming of finishing this thesis was a dream.

Publications

1. H. Nikmehr and C. C. Lim. A New On-the-fly Summation Algorithm. In *Proceedings of 8th Asia-Pacific Computer Systems Architecture Conference ACSAC 2003*, volume 2823 of *Lecture Notes in Computer Science*, pages 258267, Aizu-Wakamatsu, Japan, 2326 September 2003.

2. H. Nikmehr and B. Phillips and C.C. Lim. A Decimal Carry-Free Adder. In *Proceeding of SPIE conference on Smart Materials, Nano-, and Micro-Smart Systems 2004*, pages 786-797, Sydney, Australia, 13-15 December 2004.

List of Principal Symbols

$1.f$	significand or mantissa (IEEE 754 standard)
a	largest digit in a SD set
b	borrow
B_i	group borrow
β	representation radix (IEEE 754 standard)
β_P	number of integer bits of the shifted PR (PD plot)
C_{in_i}	gate i input capacitance (logical effort)
C_{out_i}	load capacitance driven by the gate i (logical effort)
c'	number of fractional digits of the shifted PR used in the comparisons
c''	number of fractional digits of the shifted PR used in the PR sign detection
d	divisor significand or *coefficient*
D	d (PD plot)
D	divisor
$\widehat{D}$	minimum path delay (logical effort)
E	exponent (IEEE 754 standard)
e'	number of shifted PR integer digits used in the comparisons
e''	number of shifted PR integer digits used in the PR sign detection
ε_D	precision at which D is examined (PD plot)
ε_P	precision at which P is examined (PD plot)
ε_q	error of q with respect to an infinite precision quotient $\frac{x}{d}$
$\varepsilon_q[j+1]$	error of q after $(j+1)$-th iteration
e_D	divisor exponent
e_Q	quotient exponent
e_X	dividend exponent
f_i	stage effort (logical effort)
$\widehat{f}$	minimum stage effort (logical effort)

F	path effort (logical effort)
f	a dimension related to the fan out (parallel-prefix taxonomy)
g_i	stage logical effort (logical effort)
G	path logical effort (logical effort)
g	generate
G	guard bit (rounding)
h_i	stage electrical effort (logical effort)
$\widehat{h_i}$	minimum stage electrical effort (logical effort)
H	path electrical effort (logical effort)
k	kill
l	dimension related to number of logic levels (parallel-prefix taxonomy)
LE_i	total logical effort born by a component
L	last bit (rounding)
L_k	continuity condition lower bound
$Mag(k)$	magnitude of SD number k
$m.n$	number represented in m integer and n fractional digits/bits
m_k	selection constant
M_k	comparison multiple
$\widehat{N}$	best number of stages (logical effort)
N_D	total number of bits of D, which are examined (PD plot)
N_P	total number of bits of P, which are examined (PD plot)
O	proportional to
$P_{i:j}$	group propagate
p	propagate
P	PR (PD plot)
p	precision (DFP)
p_i	stage parasitic delay (logical effort)
P	path parasitic delay (logical effort)
q	quotient significand or *coefficient*
Q	quotient
$q[j+1]$	q after $(j+1)$-th iteration
QDS^*	QDS function without the PR sign detector
$q_{H_{j+1}}$	the most significant part of a radix-16 quotient digit
q_{j+1}	$(j+1)$-th quotient digit

Q_{j+1}	q in the 2's complement representation, after q_{j+1} is selected
$q_{L_{j+1}}$	least significant part of a radix-16 quotient digit
r	radix
R	round bit (rounding)
rem	remainder
ρ	redundancy factor
S	sign (IEEE 754 standard)
S	sticky bit (rounding)
s_D	dividend sign
$Sign(k)$	sign of SD number k
s_k	separating point
S_k	sign of SD number k
s_Q	quotient sign
$S_{rw[j]}$	shifted PR sign
s_X	dividend sign
t	dimension related to number of wiring tracks (parallel-prefix taxonomy)
τ	inverter delay with an identical inverter as the load
t_{in}	transfer digit from the adjacent addition position in the right
t_{out}	transfer digit sent to the adjacent addition position in the left
U_k	continuity condition upper bound
ulp	unit of last position
$w[j+1]$	$(j+1)$-th PR
w_{INT}	intermediate PR (radix-16 division)
x	dividend significand or *coefficient*
X	dividend
$<y>$	y-th digit/bit
$\lfloor y \rfloor$	the largest integer smaller than or equal to y
$\lceil y \rceil$	the smallest integer larger than or equal to y
$\neg y$	inverted y (y is a bit vector)
$\overline{y}$	$-y$ (y is a digit)
z^{C}	9's complemented z
lsf	digits involved in the least significant formation (radix-16 division)
$\{z\}_x$	number z truncated to x fractional digits/bits

List of Abbreviations

BB-DCFA	A DCFA with BCD addend and augend
BS	Borrow Save
BSD	Binary SD
BUF	Buffer
CAD	Computer Aided Design
CFA	Carry-Free Adder
CLA	Carry Look Ahead
CMOS	Complementary Metal Oxide Semiconductor
CR	Convert and Round
CRN	Convert, Round and Normalise
CS	Carry-Save
DB-DCFA	A DCFA with DSD addend and BCD augend
DCFA	Decimal Carry-Free Adder
DD-DCFA	A DCFA with DSD addend and augend
DFP	Decimal Floating-Point
DSD	Decimal Signed-Digit
DSD2BCD	DSD to BCD
DSP	Digital Signal Processor
EDA	Electronic Design Automation
FP	Floating-Point
FPU	FP Unit
FRFU	Final Result Formation Unit
FRSU	Final Result Selection Unit
IEEE	Institute of Electrical and Electronics Engineers
MRC	Multilevel Reverse-Carry
MUX	Multiplexer

PLA	Programmable Logic Array
PR	Partial Remainder
QDS	Quotient Digit Selection
RHE	Round-Half-Even
RTL	Register Transfer Language
RTNE	Round To Nearest Even
SD	Signed-Digit
SRT	Type of non-restoring digit recurrence division named after D. Sweeney, J. E. Robertson and T. D. Tocher
TBSU	Transfer Bit Selection Unit
TDSU	Transfer Digit Selection Unit
VLSI	Very Large Scale Integration