
Floating point

-Floating point implement in FPGA-

By Van Duc NGUYEN

Candidate 20151050

Ngày 10 tháng 12 năm 2019

Mục lục

Danh sách hình vẽ

Danh sách bảng

Chương 1

Introduction

Ngày nay,trong các thế hệ chip xử lý việc tính toán hiệu năng cao là 1 yêu cầu cần thiết.Để thực hiện nhiệm vụ đó,rất nhiều hàm tính toán số học được thực hiện bên bộ xử lý của phần cứng nhanh hơn so với tính toán trên phần mềm trong cùng một nhiệm vụ.Những hoạt động chính của quá trình xử lý là tính toán các hàm số học,được ứng dụng rất rộng rãi trong tính toán khoa học,xủ lý ảnh và tín hiệu..Đăc biệt trong quá trình xử lý tín hiệu,các hàm tính toán cơ bản như cộng,nhân,chia được sử dụng rất rộng rãi trong nhiều ứng dụng.

Trong biểu diễn số học trong máy tính có 2 kiểu biểu diễn phổ biến,đó là kiểu fixed-point và kiểu floating-point.Với kiểu biểu diễn fixed-point(ví dụ:mã bù 2),hoàn toàn có thể biểu diễn các số nguyên dương và âm hoặc các số gần 0.Bằng cách cố định radix 2 hoặc radix cao hơn.Với kiểu định dạng này,có thể biểu diễn số với phần thập phân 1 cách dễ dàng.

Tuy nhiên,cách tiếp cận này có vài hạn chế.Không thể biểu diễn các số rất lớn hoặc các số rất nhỏ.Hơn nữa,phần thập phân của thương của 1 phép chia 2 số có thể bị mất mát. Ví dụ với hệ thập phân,với số 976,000,000,000,000 có thể biểu diễn thành $9.76 * 10^{14}$,bằng cách biểu diễn như vậy, có thể dễ dàng biểu diễn các giá trị rất lớn hoặc rất nhỏ trong các bộ nhớ có giới hạn trong các phần cứng.Việc sử dụng floating point giúp có thể tính toán trên dải rộng và độ chính xác cao.

Công việc trong báo cáo này bao gồm thiết kế và kiểm tra các phép toán số học cơ bản:cộng,nhân,chia,phép nghịch đảo và căn bậc 2 trên dấu phảy động.Cả đầu vào và đầu ra của các phép toán đều được thiết kế cho dấu phảy động với độ chính xác đơn theo tiêu chuẩn của IEEE 754/2008 cho dấu phảy động.

Chương 2

Floating-point representation

2.1 Biểu diễn số floating-point

Biểu thức biểu diễn dạng của số floating-point

$$\pm S * B^{\pm E}$$

1 số bất kỳ được biểu diễn dưới dạng cơ số B: nhị phân(hoặc thập phân) với 3 thành phần:

- Sign:phần dấu(cộng(+) hoặc trừ(-))
- Significand S :phần định trị(còn có tên khác là mantissa hoặc fraction)
- Exponent E : phần mũ.

Hình 2.1: Định dạng của số 32-bit floating-point

Trong công thức trên cơ số B được ẩn và không cần phải lưu trữ bởi vì nó dùng để biểu diễn cho toàn bộ số.Hình 2.1 biểu diễn định dạng của số 32-bit floating-point dạng cơ số 2,trong đó giá trị bit lớn nhất dùng để lưu trữ dấu của số cần biểu diễn(0= dương,1= âm).Giá trị phần mũ được lưu vào 8 bit tiếp theo,đây là kiểu biểu diễn BRN(biased representation number) với giá trị 1 giá trị cố định được gọi là bias,khi tính giá trị thật của số mũ cần biểu diễn ta lấy giá trị BRN trừ đi bias.Giá trị bias được tính theo công thức $(2^{k-1}-1)$,với k là số bit mà phần mũ biểu diễn dưới dạng binary(trong trường hợp này là 8 bit).Do đó dải sô mũ biểu diễn từ 0-255 tương ứng với dải biểu diễn của số mũ thật từ -127 đến +128.Phần bit còn lại(trong trường hợp này là 23 bit cuối) là phần định trị(significand).

2.1.1 Normalized,unnormalized and denormalized representation

Số floating-point có nhiều cách biểu diễn khác nhau.ví dụ các giá trị dưới đây đều biểu diễn 1 số,nhưng số mũ và phần định trị khác nhau:

$$0.1101 * 2^6$$

$$1101 * 2^2$$

$$0.01101 * 2^7$$

Dạng chuẩn hóa(Normalized)

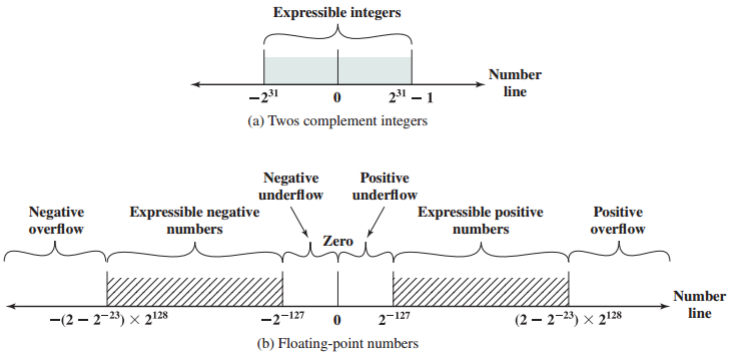
Để đơn giản trong việc tính toán trên floating-point.Các số cần được chuẩn hóa về dạng:

$$\pm 1.mmm...m * B^{\pm E}$$

Với cơ số 2,biểu thức biểu diễn dạng chuẩn hóa là:

$$\pm 1.bbb...b * 2^{\pm E}$$

Với b là số binary(0 hoặc 1).Bởi vì MSB(most significant bit) luôn là 1 nên không cần lưu trữ và được coi là bit ẩn(hidden bit).Vì vậy giá trị của phần định trị ở dạng chuẩn hóa nằm trong khoảng từ [1,2).
Hình 2.2 so sánh dải biểu diễn của kiểu fixed-point và floating point với cùng độ dài 32-bit.



Hình 2.2: Dải biểu diễn của fixed-point và floating trong định dạng 32-bit

Với hình 2.2.a sử dụng kiểu biểu diễn mã bù 2(thuộc kiểu fixed-point),với dải biểu diễn từ -2^{31} đến $2^{31}-1$ và tổng số biểu diễn được là 2^{23}.Đối với số floating-point,dải biểu diễn được thể hiện trong hình 2.2.b:

- số âm nằm trong khoảng từ $-(2-2^{-23})*2^{128}$ đến -2^{-127}
- số dương nằm trong khoảng từ 2^{-127}đến $(2-2^{-23})*2^{128}$

Trong đó có 5 vùng không thuộc khoảng trên:

- các số âm nhỏ hơn $-(2-2^{-23})*2^{128}$ được gọi là negative overflow.
- các số âm lớn hơn -2^{-127} được gọi là negative underflow.
- số 0
- các số dương nhỏ hơn 2^{-127} được gọi là positive underflow.
- các số dương lớn hơn $(2-2^{-23})*2^{128}$ được gọi là positive overflow.

Các số thuộc vùng negative underflow và positive underflow được gọi là unnormalized.

Giá trị **0** là giá trị đặc biệt mà kiểu biểu diễn sử dụng với phần định trị bằng 0 và phần mũ là giá trị nhỏ nhất của số mũ.

Dạng chưa chuẩn hóa(Denormalized)

Các số unnormalized được được chuẩn IEEE 754 quy định là số denormalized.Biểu thức của dạng biểu diễn denormalized:

$$\pm 0.mmm...m * B^{E_{min}}$$

Trong đó phần hidden bit chuyển từ **1.** sang **0.** .Và số mũ chỉ cho phép là số mũ nhỏ nhất.

2.1.2 Biểu diễn phần định trị(Significand)

Phần định trị là 1 số có dấu.Có 2 cách phổ biến để biểu diễn 1 số có dấu là sign-and-magnitude và 2's complement(mã bù 2).Hiện nay các biểu diễn sign-and-magnitude được dùng nhiều hơn vì nó biểu diễn số một cách tự nhiên và đơn giản khi thực hiện các hàm nhân,chia trên dấu phảy động.Với cơ số B=2,trong dạng chuẩn hóa bit lớn nhất của significand luôn là 1.Vì vậy nó không cần thiết được lưu trữ trong định dạng của kiểu biểu diễn.MSB của phần significand được gọi là **hidden bit**

2.1.3 Biểu diễn phần mũ(Exponent)

Phần mũ E là kiểu số nguyên có dấu,có thể biểu diễn bằng nhiều cách như sign-and-magnitude, true-and-comlement,và biased.Cách biểu diễn biased được chọn bởi vì nó rất đơn giản khi so sánh 2 số floating-point và giá trị nhỏ nhất của số mũ là 0,vì vậy có thể biểu diễn giá trị 0 trong floating point. Giá trị biểu diễn biased với bias là B,giá trị số mũ E được biểu diễn là số nguyên dương E_R:

$$E_R = E + B$$

Giá trị nhỏ nhất của số mũ E_R=0,nên:

$$B = -E_{min}$$

Dựa vào tính đối xứng của dải biểu diễn số mũ ta có:

$$-B \leq E \leq B$$

Suy ra

$$0 \leq E_R \leq 2B$$

Với k là sô bit của kiểu biểu diễn binary của E_R:

$$2B \leq 2^k - 1$$

Cuối cùng ta được:

$$B \leq \frac{1}{2}(2^k - 2) = 2^{k-1} - 1$$

2.1.4 Các giá trị đặc biệt

Có một vài giá trị không biểu diễn được trong hệ thống floating-point nhưng nó rất cần thiết.Ví dụ như số vô cùng(infinity) hay NaN(Not a Number).Có nhiều phép toán mà khi thực hiện giá trị trả về không phải số ví dụ như phép căn bậc hai của một số âm,số vô cùng chia cho 0,hoặc kết quả ra là số vô cùng như phép chia cho 0...

2.2 Các chế độ làm tròn và phân tích lỗi

Có 4 chế độ làm tròn(Rounding) được IEEE quy định trong floating-point

Chương 3

IEEE 754/2008 standard for binary floating-point

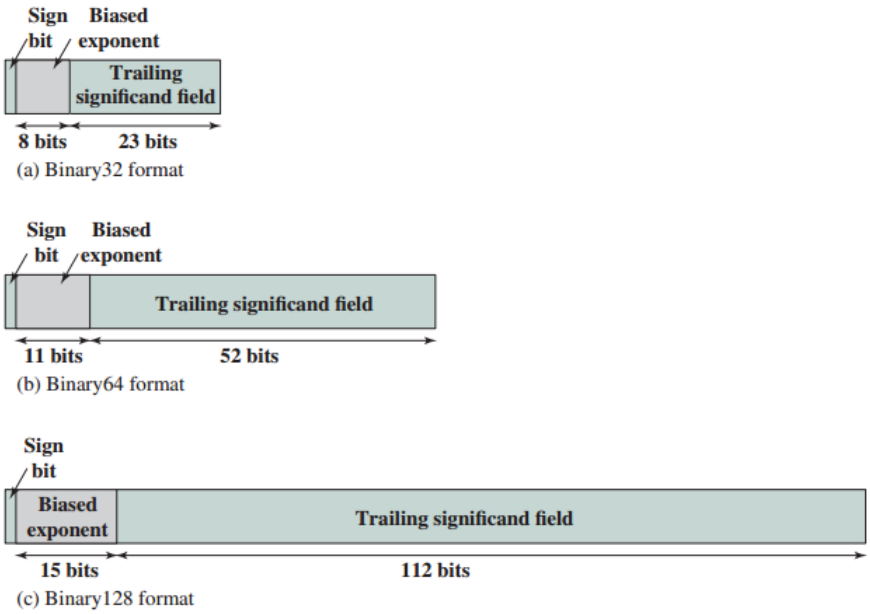
Trong tiêu chuẩn IEEE 754/2008 có 2 tiêu chuẩn biểu diễn số floating-point dưới dạng cơ số 2 và cơ số 10.Trong báo cáo này đề cập tới việc biểu diễn số floating-point với cơ số 2.Tiêu chuẩn này được phát triển giúp cho sự chuyển đổi linh hoạt các chương trình từ một bộ xử lý này sang các bộ xử lý khác và được khuyên dùng trong pháp triển các chương trình tinh vi,định hướng số. IEEE 754/2008 định ngĩa 3 kiểu định dạng floating-point khác nhau.

- Arithmetic format(Định dạng số học):Định dạng này có thể sử dụng để thể hiện biểu diễn dấu phảy động với các toán hạng hoặc kết quả sau khi thực hiện các hàm tính toán được mô tả trong tiêu chuẩn.
- Basic format(Định dạng cơ bản):Định dạng này bao gồm 5 kiểu kiểu diễn dấu phảy động(3 kiểu nhị phân và 2 kiểu cho thập phân).Và được mã hóa theo tiêu chuẩn ,được sử dụng cho tính toán số học.
- Interchange format(Định dạng trao đổi):Mã hóa nhị phân có độ dài cố định,đầy đủ cho phép trao đổi dữ liệu giữa các nền tảng khác nhau và có thể được sử dụng để lưu trữ.

3.1 Tiêu chuẩn IEEE 754/2008 cho floating-point binary.

Có 3 định dạng cơ bản của dấu phảy động nhị phân với chiều dài bit là 32,64 và 128 bit,tương ứng với độ rộng của số bit lần lượt là 8 bit,11 bit và 15 bit.(2 định dạng cơ bản còn lại cho dấu phảy động thập phân với độ dài bit là 64 và 128 bit).Tất cả các định dạng cơ bản cũng đồng thời là định dạng số học(được sử dụng cho tính toán số học) và định dạng trao đổi. Ngoài ra còn có vài định dạng cho ứng dụng chuyên biệt như định dạng 16 bit nhi phân khi tính toán

không yêu cầu độ chính xác cao.Hoặc định dạng với độ chính xác mở rộng dành cho ứng dụng yêu cầu độ chính xác cao.



Hình 3.1: Định dạng IEEE 754

Hình 3.1 thể hiện định dạng IEEE 754 cho độ dài bit lần lượt là 16 bit(single precision),32bit(double precision),128 bit.

Parameter	Format		
	Binary32	Binary64	Binary128
Storage width (bits)	32	64	128
Exponent width (bits)	8	11	15
Exponent bias	127	1023	16383
Maximum exponent	127	1023	16383
Minimum exponent	−126	−1022	−16382
Approx normal number range (base 10)	$10^{-38}, 10^{+38}$	$10^{-308}, 10^{+308}$	$10^{-4932}, 10^{+4932}$
Trailing significand width (bits)*	23	52	112
Number of exponents	254	2046	32766
Number of fractions	2^{23}	2^{52}	2^{112}
Number of values	1.98×2^{31}	1.99×2^{63}	1.99×2^{128}
Smallest positive normal number	2^{-126}	2^{-1022}	2^{-16362}
Largest positive normal number	$2^{128} - 2^{104}$	$2^{1024} - 2^{971}$	$2^{16384} - 2^{16271}$
Smallest subnormal magnitude	2^{-149}	2^{-1074}	2^{-16494}

Hình 3.2: Thông số của định dạng IEEE 754

Hình 3.2 thể hiện mối quan hệ giữa cách kiểu định dạng với độ dài bit khác nhau.

Không phải tất cả các mẫu bit trong định dạng của IEEE 754 đều được giải thích theo cách thông thường,thay vào đó có một vài mẫu bit được sử dụng để biểu diễn giá trị đặc biệt.

(a) binary32 format

	Sign	Biased Exponent	Fraction	Value
positive zero	0	0	0	0
negative zero	1	0	0	−0
plus infinity	0	all 1s	0	∞
minus infinity	1	all 1s	0	−∞
quiet NaN	0 or 1	all 1s	≠ 0; first bit = 1	qNaN
signaling NaN	0 or 1	all 1s	≠ 0; first bit = 0	sNaN
positive normal nonzero	0	0 < e < 225	f	$2^{e-127}(1.f)$
negative normal nonzero	1	0 < e < 225	f	$-2^{e-127}(1.f)$
positive subnormal	0	0	f ≠ 0	$2^{e-126}(0.f)$
negative subnormal	1	0	f ≠ 0	$-2^{e-126}(0.f)$

(b) binary64 format

	Sign	Biased Exponent	Fraction	Value
positive zero	0	0	0	0
negative zero	1	0	0	−0
plus infinity	0	all 1s	0	∞
minus infinity	1	all 1s	0	−∞
quiet NaN	0 or 1	all 1s	≠ 0; first bit = 1	qNaN
signaling NaN	0 or 1	all 1s	≠ 0; first bit = 0	sNaN
positive normal nonzero	0	0 < e < 2047	f	$2^{e-1023}(1.f)$
negative normal nonzero	1	0 < e < 2047	f	$-2^{e-1023}(1.f)$
positive subnormal	0	0	f ≠ 0	$2^{e-1022}(0.f)$
negative subnormal	1	0	f ≠ 0	$-2^{e-1022}(0.f)$

(c) binary128 format

	Sign	Biased Exponent	Fraction	Value
positive zero	0	0	0	0
negative zero	1	0	0	−0
plus infinity	0	all 1s	0	∞
minus infinity	1	all 1s	0	−∞
quiet NaN	0 or 1	all 1s	≠ 0; first bit = 1	qNaN
signaling NaN	0 or 1	all 1s	≠ 0; first bit = 0	sNaN
positive normal nonzero	0	all 1s	f	$2^{e-16383}(1.f)$
negative normal nonzero	1	all 1s	f	$-2^{e-16383}(1.f)$
positive subnormal	0	0	f ≠ 0	$2^{e-16383}(0.f)$
negative subnormal	1	0	f ≠ 0	$-2^{e-16383}(0.f)$

Hình 3.3: Các trường hợp biểu diễn trong IEEE 754 floating-point

Hình 3.3 thể hiện sự phân chia thành các lớp mà số floating-point biểu diễn:

- Để biểu diễn số chuẩn hóa(normailzied),giá trị của số mũ E trong dải (0,255)đối với định dạng 32-bit,đối với 64-bit là từ $(0, 2047)$và nằm trong dải(0,16383) đối với định dạng 64-bit.Giá trị thực của số mũ đã được hiệu chỉnh lệch đi một lượng bias,ví dụ dải thực của số mũ trong định dạng 32-bit là $[-126, 127]$.Đồng thời khi ở dạng normalized yêu cầu bit **1** ở bên trái binary point(**1.**).Bit này được ẩn và là bit MSB(most significand bit) đối với phần định trị(mantissa hay fraction).

- Để biểu diễn số **0**,phần mũ E=0 và phần định trị F=0,dấu ±0 được quyết định bởi bit dấu(sign bit).
- Để biểu diễn số ∞ thì tất cả bit của phần mũ E đều bằng 1(Max E) và phần định trị(fraction)F=0.Dấu ±∞ phụ thuộc vào bit dấu S.
- Để biểu diễn số chưa được chuẩn hóa(denormalized) thì phần mũ E=0 và phần định trị F $\neq$ 0.Trong trường hợp này số mũ thực mà số floating-point biểu diễn là -126(đối với 32-bit) và -1022(đối với 64-bit).Dấu của số phụ thuộc vào bit dấu S.
- Đối với giá trị NaN(Not a Number),tất cả các bit của phần mũ E đều bằng 1(Max E) và phần định trị F$\neq$ 0.

3.2 Các chế độ làm tròn trong tiêu chuẩn IEEE 754/2008 cho floating-point bianry

Khi thực hiện các phép toán số học,kết quả số có thể được lưu trữ,bị dịch đi..Khi đưa về định dạng của số floating-point có thể bị cắt bỏ hoặc làm tròn tới giá trị chính xác gần nhất.Tiêu chuẩn của IEEE đưa ra 4 chế độ làm tròn:

- Round to nearnest:làm tròn về số gần nhất.
- Round toward $+\infty$:làm tròn lên(tiến tới dương vô cùng).
- Round toward $-\infty$:làm tròn xuống(tiến tới âm vô cùng).
- Round toward 0:làm tròn về gần 0.

Trong đó,chế độ làm tròn mặc địch được quy định là Round to nearnest.Kết quả sau khi được chuẩn hóa(normalized) có dạng $\mathbf{1.}f_1f_2...f_l\mathbf{RS}$

- f_l:(last fraction bit)là bit cuối cùng của phần định trị theo định dạng(bit thứ 23 đối với định dạng floating-point 32-bit).
- R:(round bit)là bit đầu tiên sau bit f_l.
- S:(sticky bit)xuất hiện sau round bit(được tính bằng phép OR các bit còn lại).

4 trường hợp của **RS**:

1. **RS**=00:Kết quả đã chính xác,không cần làm tròn.
2. **RS**=01:làm tròn kết quả xuống bằng cách cắt bỏ phần RS.

3. **RS**=11:làm tròn kết quả lên bằng cách cộng thêm 1 vào f_l.

4. **RS**=10:tăng hoặc giảm phụ thuộc vào f_l

    - f_l=0:làm tròn kết quả xuống bằng cách cắt bỏ phần RS.
    - f_l=1:làm tròn kết quả lên bằng cách cộng thêm 1 vào f_l.

Chương 4

Floating-point arithmetic

4.1 Addition and Subtraction

Phép cộng và phép trừ trong floating-point có chunng cách thực hiện(khi thực hiện phép trừ chỉ cần đảo dấu của 1 toán hạng và sử dụng phép cộng). Với đầu vào là 2 số floating-point X(S_X, E_X, F_X)và Y(S_Y, E_Y, F_Y).Kết quả sau khi thực hiện phép toán X+Y là Z(S_Z, E_Z, F_Z).
Các bước thực hiện phép toán cộng/trừ dấu phảy động:

1. Tách đầu vào thành 3 phần:phần dấu(sign),phần mũ(expoent) và phần định trị(mantissa).Kiểm tra đầu vào xem có thuộc các trường hợp đặc biệt như là số vô cùng,là số 0 hoặc số không hợp lệ theo chuẩn biểu diễn IEEE 754.Thêm bit ẩn vào phần định trị.

2. So sánh sô mũ của 2 số đầu vào:số nhỏ hơn sẽ bị dịch phải phần định trị để 2 số có cùng số mũ.Số bit dịch của phần định trị phụ thuộc vào sự sai khác giữa 2 số mũ.Sau đó cộng hoặc trừ phần định trị mới với mantissa của số lớn hơn.

3. Thực hiện phép dịch trái phần mantissa cho tới khi bit MSB bằng 1.Với mỗi lần dịch,giảm giá trị phần mũ đi 1.Làm tròn phần định trị nếu cần thiết.Cuối cùng đóng gói phần dấu,phần mũ và phần định trị theo định dạng và đưa ra kết quả

Mã giả(pescode) của phép toán cộng 2 số floating-point.
bảng thể hiện giá trị phần dấu,mũ và mantissa của đầu ra theo đầu vào.
Sơ đồ khối thuật toán cộng 2 số dấu phảy động.

4.2 Multiplication and Divison

4.3 Reciprocal and Square Root

Chương 5

Implement in FPGA

5.1 Addition and Subtraction

5.2 Multiplication and Division

5.3 Reciprocal and Square Root

Chương 6

Evaluation

Chương 7

Conclussion and Related-work

This is end of document

[1]

Tài liệu tham khảo

[1] Duc Nguyen,"sach cua toi",*bai bao* Sun,25,May,2019.

[2] Authors *"title of article"* Title of Journal Date,Month,Year page Medium of publish