معماری کامپیوتر (۱۳۹۱-۱۱-۱۳)
جلسه‌ی یازدهم

دانشگاه شهید بهشتی
دانشکده‌ی مهندسی برق و کامپیوتر
بهار ۱۳۹۱
احمد محمودی ازناوه

فهرست مطالب

- مروری بر جلسه‌ی پیش
- ممیز شناور

ممیز شناور

- برای نمایش اعداد اعشاری و اعداد بسیار بزرگ از سیستم عددی میز شناور استفاده می شود.
  - ۳,۱۴۱۵۹۲۶۵
  - ۲,۷۱۸۲۸
  - ۰۰۰۰۰۰۰۰۱ = ۰,۱× ۱۰$^{-۹}$

	IBM/370	DEC/VAX	Cyber 70
Word length (double)	32 (64) bits	32 (64) bits	60 bits
Significand+{hidden bit}	24 (56) bits	$23+1$ $(55+1)$ bits	48 bits
Exponent	7 bits	8 bits	11 bits
Bias	64	128	1024
Base	16	2	2
Range of M	$\frac{1}{16} \le M < 1$	$\frac{1}{2} \le M < 1$	$1 \le M < 2$
Representation of M	Signed-magnitude	Signed-magnitude	One's complement
Approximate range	$16^{63} \approx 7 \cdot 10^{75}$	$2^{127} \approx 1.9 \cdot 10^{38}$	$2^{1023} \approx 10^{307}$
Approximate resolution	$2^{-24} \approx 10^{-7}$ (10^{-17})	$2^{-24} \approx 10^{-7}$ (10^{-17})	$2^{-48} \approx 10^{-14}$

Exponential Notation

- تمام اعداد زیر نمایش عدد 1234 می‌باشند.

$123{,}400.0 \times 10^{-2}$

$12{,}340.0 \times 10^{-1}$

$1{,}234.0 \times 10^{0}$

123.4×10^{1}

12.34×10^{2}

1.234×10^{3}

0.1234×10^{4}

با تغییر همزمان توان و جایگاه ممیز نمایش‌های متفاوتی برای یک عدد به دست می‌آید.

ممیز شناور (ادامه...)

- در سال ۱۹۸۵ استاندارد IEEE Std 754 مطرح شد.
- این استاندارد واگرایی شیوه‌های به کار رفته برای نمایش ممیز شناور را کاهش داد.
  - بدین‌ترتیب برنامه‌های نوشته شده برای مقاصد علمی قابل حمل شدند.
- بر طبق این استاندارد، اعداد به دو شیوه نشان داده می‌شود:

- single
- double

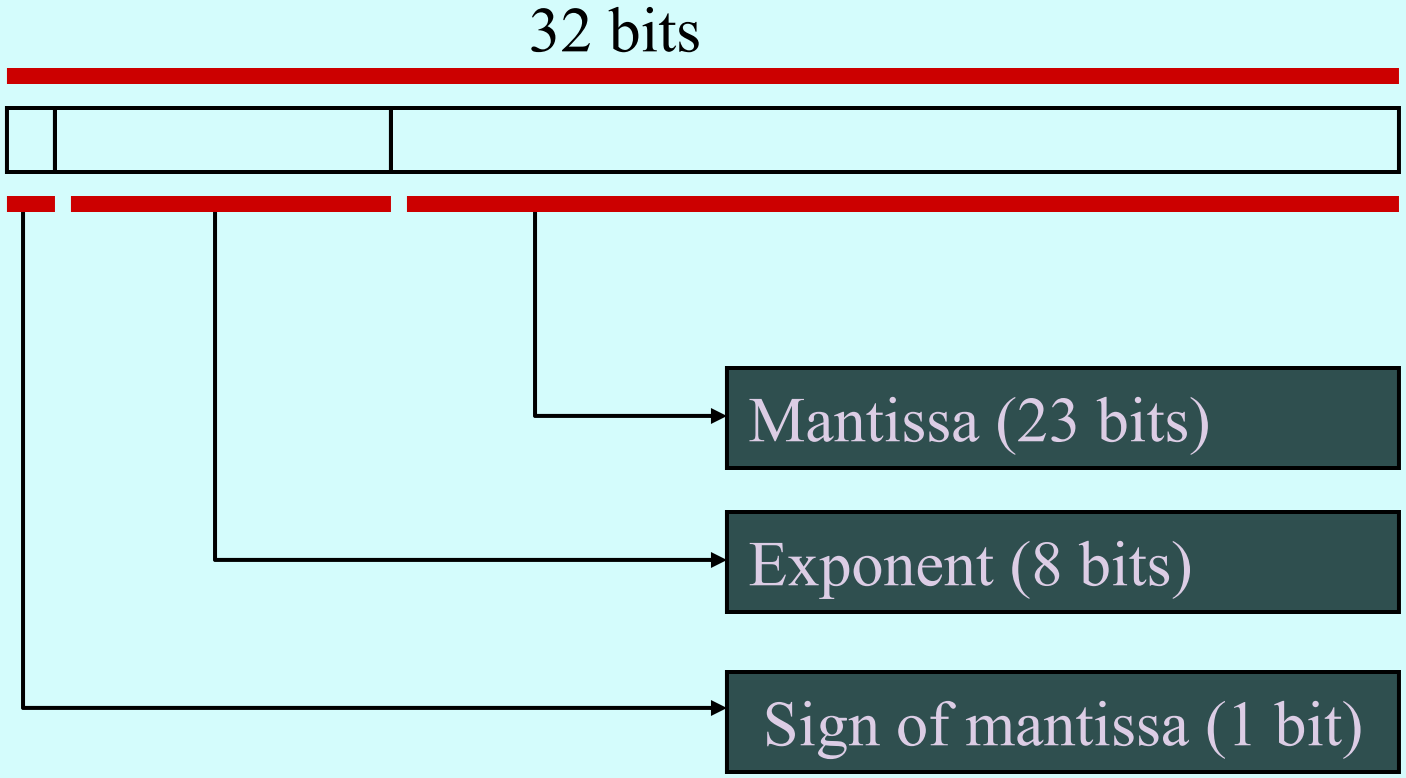
single: 8 bits
double: 11 bits

single: 23 bits
double: 52 bits

$$x = (-1)^S \times (1 + \text{Fraction}) \times 2^{(\text{Exponent} - \text{Bias})}$$

Single: Bias = 127; Double: Bias = 1023

Single Precision Format



$$Value = (-1)^S \; 1.F \times 2^{E-127}$$

Denormal Numbers

اعداد نانرمال

Exponent = 000...0 $\Rightarrow$ hidden bit is 0

$$x = (-1)^S \times (0 + \text{Fraction}) \times 2^{-\text{Bias}}$$

- بدین ترتیب می‌توان اعداد کوچک‌تری را نیز نمایش داد.
- در صورتی که بخش کسری را برابر صفر قرار دهیم:

$$x = (-1)^S \times (0 + 0) \times 2^{-\text{Bias}} = \pm 0.0$$

بدین ترتیب دو نمایش برای 0 خواهیم داشت

Infinities and NaNs

ناعدد و بی‌نهایت

- Exponent = 111...1, Fraction = 000...0
  - $\pm\infty$
    - در محاسبات بعدی نیز قابل استفاده است.
- Exponent = 111...1, Fraction ≠ 000...0
    - ناعدد (Not-a-Number (NaN))
    - بیان‌گر محاسبات نادرست می‌باشد.
    - این اعداد نیز قابلیت استفاده در محاسبات بعدی را دارند.

Single precision		Double precision		Object represented
Exponent	Fraction	Exponent	Fraction	
0	0	0	0	0
0	Nonzero	0	Nonzero	± denormalized number
1–254	Anything	1–2046	Anything	± floating-point number
255	0	2047	0	± infinity
255	Nonzero	2047	Nonzero	NaN (Not a Number)

محاسبات مقادیر خاص

- $(+0) + (+0) = (+0) - (-0) = +0$
- $(+0) \times (+5) = +0$
- $(+0) / (-5) = -0$
- $(+\infty) + (+\infty) = +\infty$
- $x - (+\infty) = -\infty$
- $(+\infty) \times x = \pm\infty$, depending on the sign of x
- $x / (+\infty) = \pm 0$, depending on the sign of x
- $\sqrt{(+\infty)} = +\infty$

نمایش در مبنای شانزده

- **نمایش یک عدد ممیز شناور در مبنای شانزده معمول است:**

0	10000011	01000000000000000000000
4	1	A	0	0	0	0	0

-15.6875

$C17B0000_{16} =$

$1\ 10000010\ 11110110000000000000000_2$

S E M

1 = negative
0 = positive

- محاسباتی که منجر به تولید ناعدد می‌شود:
  - $(\pm 0) / (\pm 0) = NaN$
  - $(+\infty) + (-\infty) = NaN$
  - $(\pm 0) \times (\pm \infty) = NaN$
  - $(\pm \infty) / (\pm \infty) = NaN$
- ناعدد در محاسبات و مقایسه‌ها
  - $NaN + x = NaN$
  - $NaN + NaN = NaN$
  - $NaN \times 0 = NaN$
  - $NaN \times NaN = NaN$

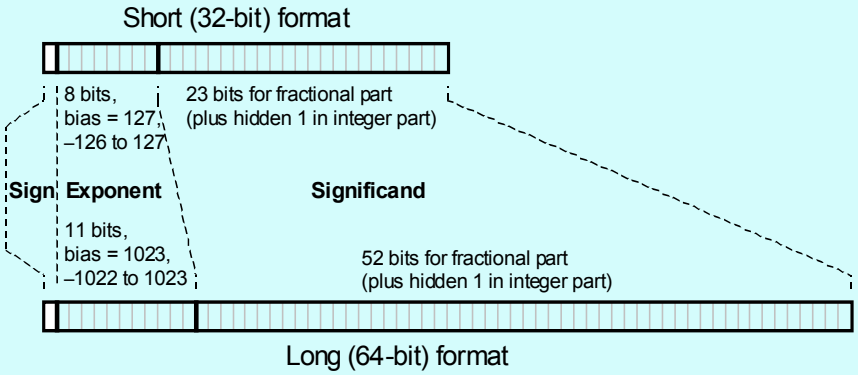
$NaN < 2 \rightarrow$ false

$NaN = Nan \rightarrow$ false

$NaN \neq (+\infty) \rightarrow$ true

$NaN \neq NaN \rightarrow$ true

پراکندگی داده‌ها در ممیز شناور

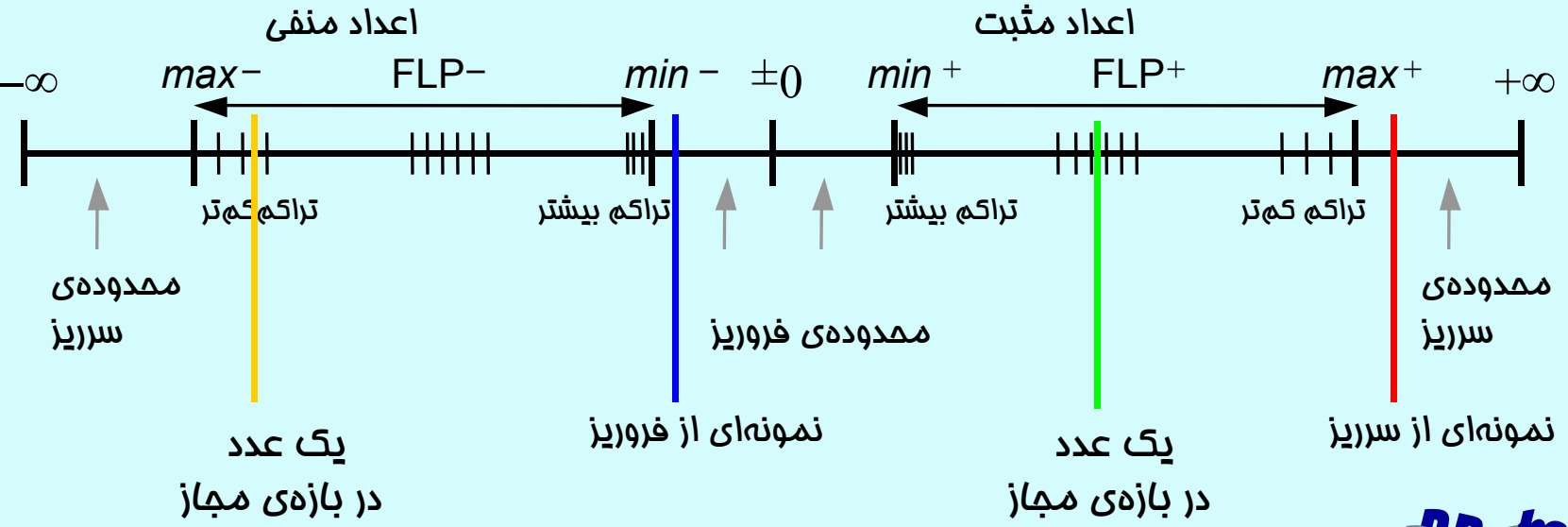


$\pm 0, \pm \infty, \text{NaN}$

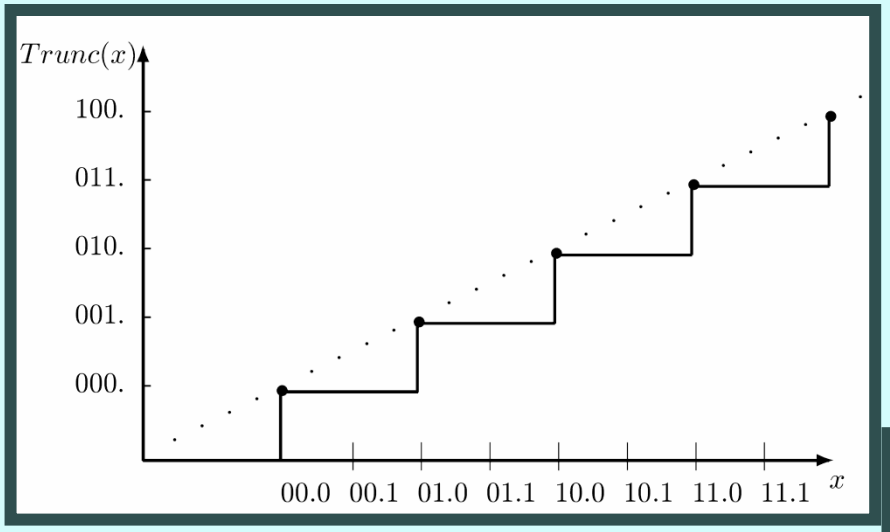
$1.f \times 2^e$

Denormals:

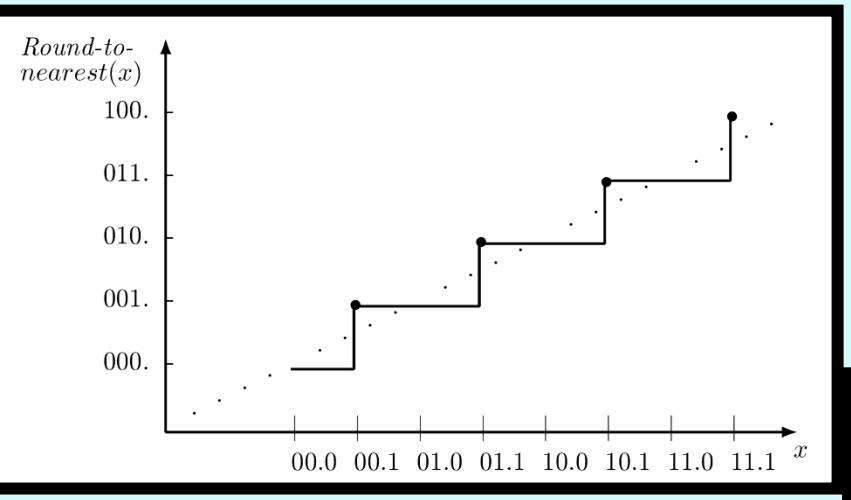
$0.f \times 2^{e_{min}}$



گرد کردن

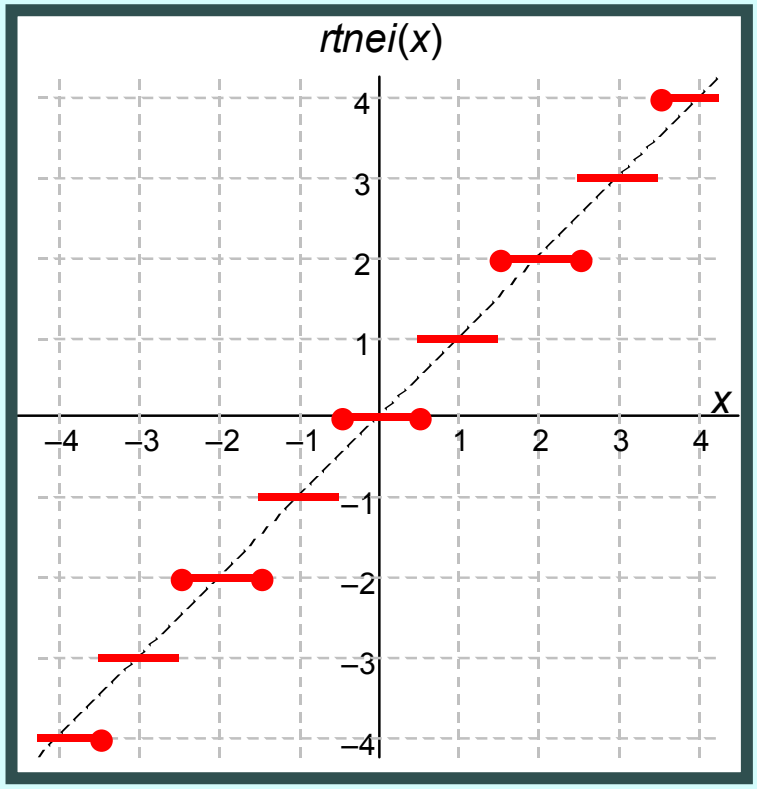


Number	$Trunc(x)$	Error
$X.00$	X	0
$X.01$	X	$-1/4$
$X.10$	X	$-1/2$
$X.11$	X	$-3/4$



Number	*Round-to-nearest(x)*	Error
$X.00$	X	0
$X.01$	X	$-1/4$
$X.10$	$X + 1$	$+1/2$
$X.11$	$X + 1$	$+1/4$

گرد کردن به نزدیک‌ترین مقدار زوج



Number	$Round(x)$	Error	Number	$Round(x)$	Error
$X0.00$	$X0.$	0	$X1.00$	$X1.$	0
$X0.01$	$X0.$	$-1/4$	$X1.01$	$X1.$	$-1/4$
$X0.10$	$X0.$	$-1/2$	$X1.10$	$X1.+1$	$+1/2$
$X0.11$	$X1.$	$+1/4$	$X1.11$	$X1.+1$	$+1/4$

شیوه‌های گرد کردن در استاندارد IEEE754

LSB	R	S	Operation	$\overline{Error}$
0	0	0	+ 0	0
0	0	1	+ 0	$-0.25\ ulp$
0	1	0	+ 0	$-0.50\ ulp$
0	1	1	$+0.5\ ulp$	$+0.25\ ulp$
1	0	0	+ 0	0
1	0	1	+ 0	$-0.25\ ulp$
1	1	0	$+0.5\ ulp$	$+0.50\ ulp$
1	1	1	$+0.5\ ulp$	$+0.25\ ulp$
			Total	0

(a) Round-to-nearest-even scheme

R	S	Operation	$\overline{Error}$
0	0	+ 0	0
0	1	+ 0	$-0.25\ ulp$
1	0	+ 0	$-0.50\ ulp$
1	1	+ 0	$-0.75\ ulp$
		Total	$-0.375\ ulp$

(b) Round-to-zero scheme

Sign	R	S	Operation
+	0	0	+ 0
+	0	1	$+1\ ulp$
+	1	0	$+1\ ulp$
+	1	1	$+1\ ulp$
−	0	0	+ 0
−	0	1	+ 0
−	1	0	+ 0
−	1	1	+ 0

(c) Round-to-plus-infinity scheme

Sign	R	S	Operation
−	0	0	+ 0
−	0	1	$+1\ ulp$
−	1	0	$+1\ ulp$
−	1	1	$+1\ ulp$
+	0	0	+ 0
+	0	1	+ 0
+	1	0	+ 0
+	1	1	+ 0

(d) Round-to-minus-infinity scheme

محدوده‌ی قابل نمایش

- **کوچک‌ترین مقدار ممکن**
  - Exponent: 00000001
    $\Rightarrow$ actual exponent = 1 – 127 = –126
  - Fraction: 000…00 $\Rightarrow$ significand = 1.0
  - $\pm 1.0 \times 2^{-126} \approx \pm 1.2 \times 10^{-38}$
- **بزرگ‌ترین مقدار ممکن**
  - exponent: 11111110
    $\Rightarrow$ actual exponent = 254 – 127 = +127
  - Fraction: 111…11 $\Rightarrow$ significand $\approx$ 2.0
  - $\pm 2.0 \times 2^{+127} \approx \pm 3.4 \times 10^{+38}$

محدوده‌ی قابل نمایش با دقت مضاعف

- کوچک‌ترین مقدار ممکن
  - Exponent: 00000000001
    $\Rightarrow$ actual exponent = 1 – 1023 = –1022
  - Fraction: 000…00 $\Rightarrow$ significand = 1.0
  - $\pm 1.0 \times 2^{-1022} \approx \pm 2.2 \times 10^{-308}$
- بزرگ‌ترین مقدار ممکن
  - Exponent: 11111111110
    $\Rightarrow$ actual exponent = 2046 – 1023 = +1023
  - Fraction: 111…11 $\Rightarrow$ significand $\approx$ 2.0
  - $\pm 2.0 \times 2^{+1023} \approx \pm 1.8 \times 10^{+308}$

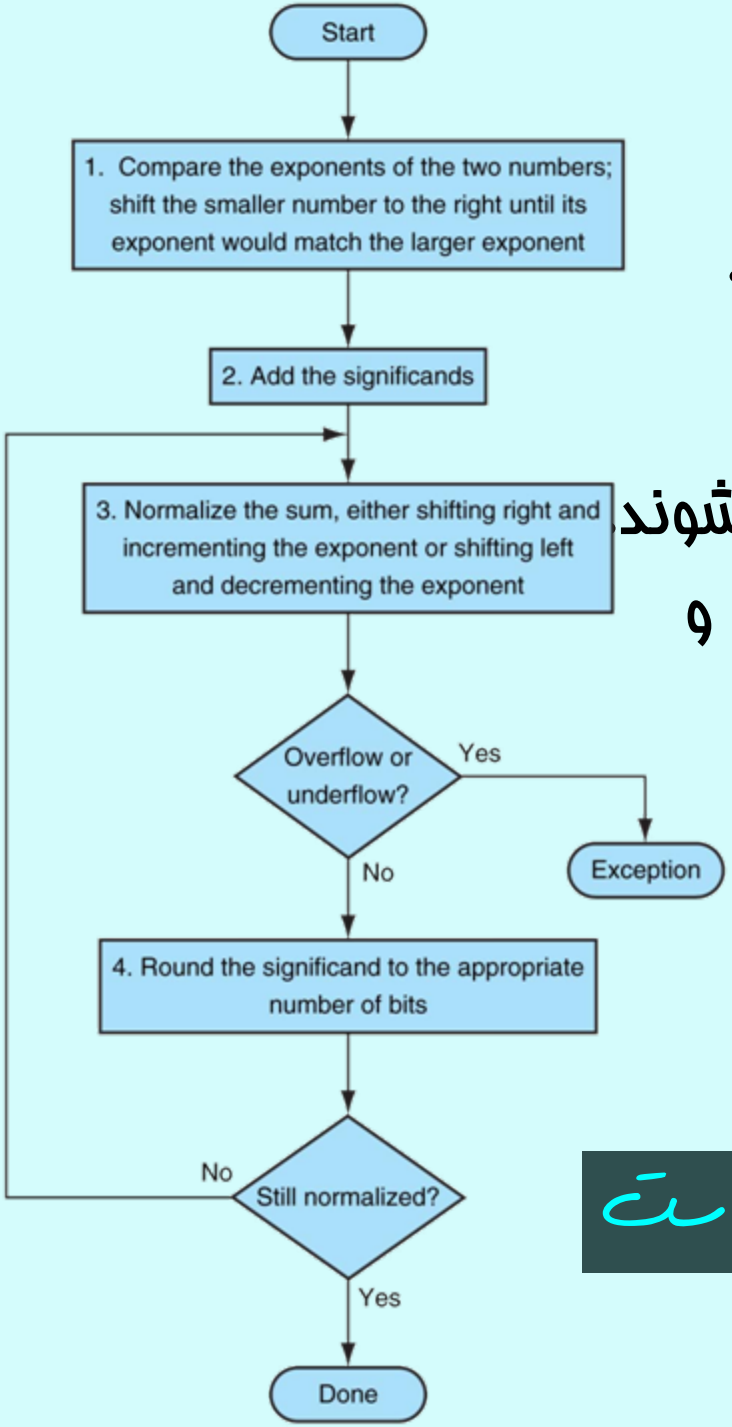
دقت در ممیز شناور

- Single: approx 2^{-23}
  - Equivalent to $23 \times \log_{10}2 \approx 23 \times 0.3 \approx 6$
  - برابر با شش رقم اعشار دقت
- Double: approx 2^{-52}
  - Equivalent to $52 \times \log_{10}2 \approx 52 \times 0.3 \approx 16$
  - برابر با شانزده رقم اعشار دقت

مشخصات اعداد ممیز شناور IEEE 754

Feature	Single/Short	Double/Long
Word width in bits	32	64
Significand in bits	23 + 1 hidden	52 + 1 hidden
Significand range	$[1, 2 - 2^{-23}]$	$[1, 2 - 2^{-52}]$
Exponent bits	8	11
Exponent bias	127	1023
Zero (± 0)	e + bias = 0, $f = 0$	e + bias = 0, $f = 0$
Denormal	e + bias = 0, $f \neq 0$ represents $\pm 0.f \times 2^{-126}$	e + bias = 0, $f \neq 0$ represents $\pm 0.f \times 2^{-1022}$
Infinity ($\pm\infty$)	e + bias = 255, $f = 0$	e + bias = 2047, $f = 0$
Not-a-number (NaN)	e + bias = 255, $f \neq 0$	e + bias = 2047, $f \neq 0$
Ordinary number	e + bias $\in [1, 254]$ $e \in [-126, 127]$ represents $1.f \times 2^e$	e + bias $\in [1, 2046]$ $e \in [-1022, 1023]$ represents $1.f \times 2^e$
min	$2^{-126} \cong 1.2 \times 10^{-38}$	$2^{-1022} \cong 2.2 \times 10^{-308}$
max	$\cong 2^{128} \cong 3.4 \times 10^{38}$	$\cong 2^{1024} \cong 1.8 \times 10^{308}$

جمع در ممیز شناور

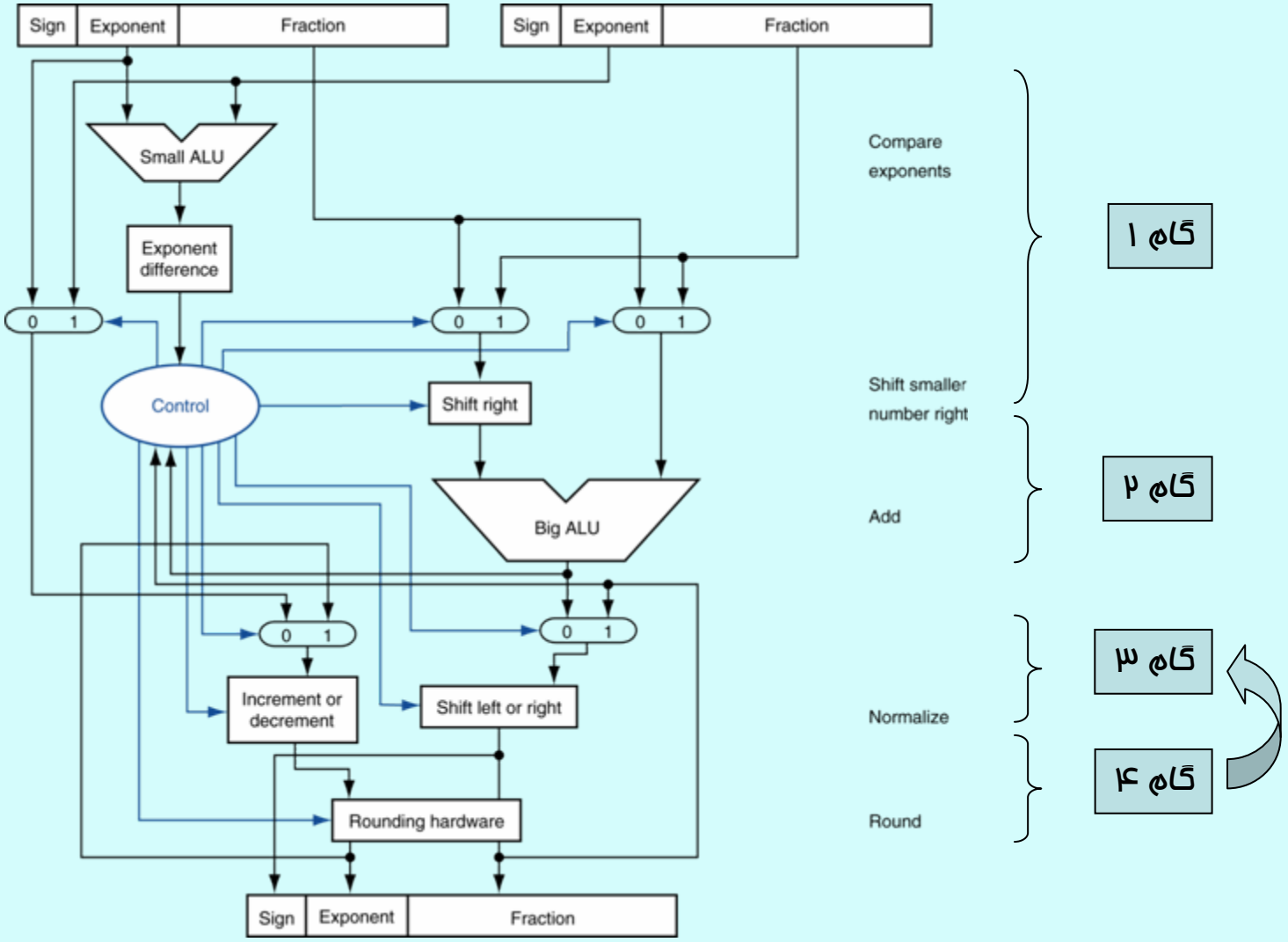


1. ابتدا توان‌ها را یکسان می‌کنیم.
   - این کار با شیفت عدد کوچک‌تر به راست انجام می‌شود.
2. مقادیر اعشاری با هم جمع می‌شوند.
3. حاصل بهنجار شده و وقوع سرریز و فروریز بررسی می‌شود.
4. حاصل گرد می‌شود.

- مجددا بهنجار بودن عدد بررسی می‌شود.

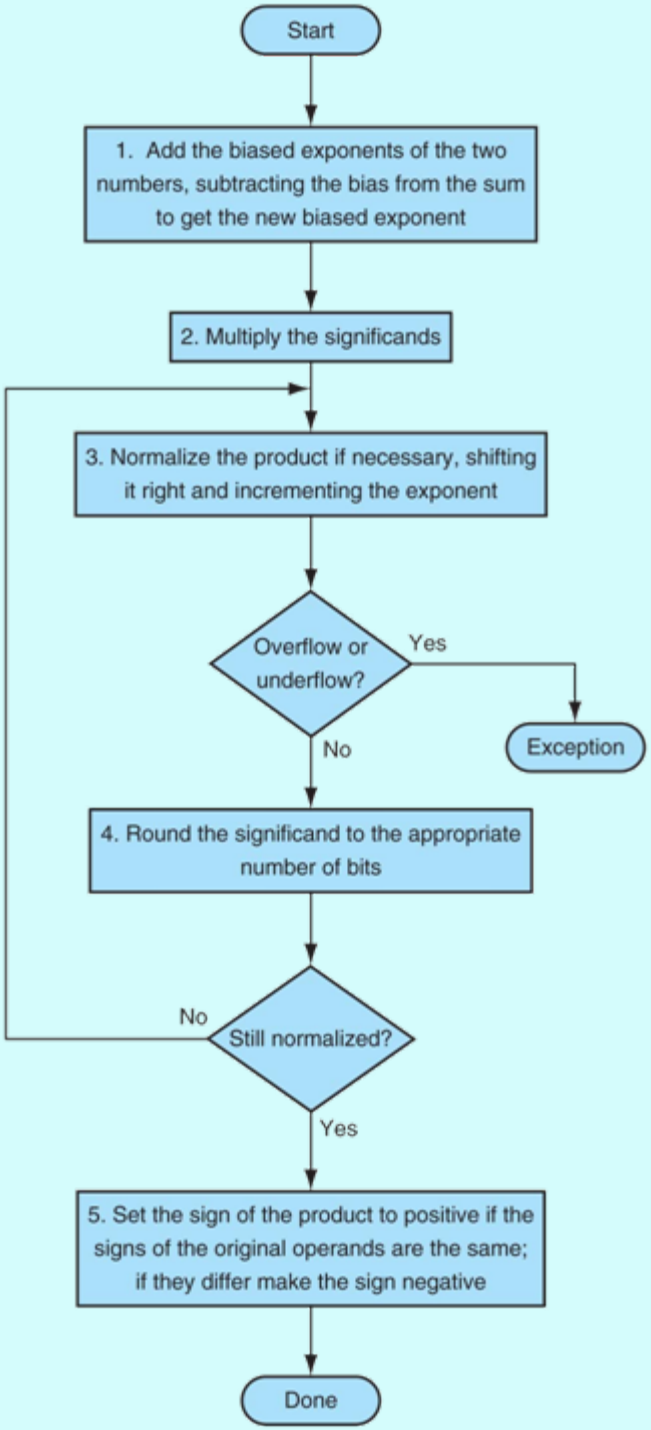
نسبت به اعداد صحیح پیچیده‌تر است

سخت‌افزار جمع ممیز شناور



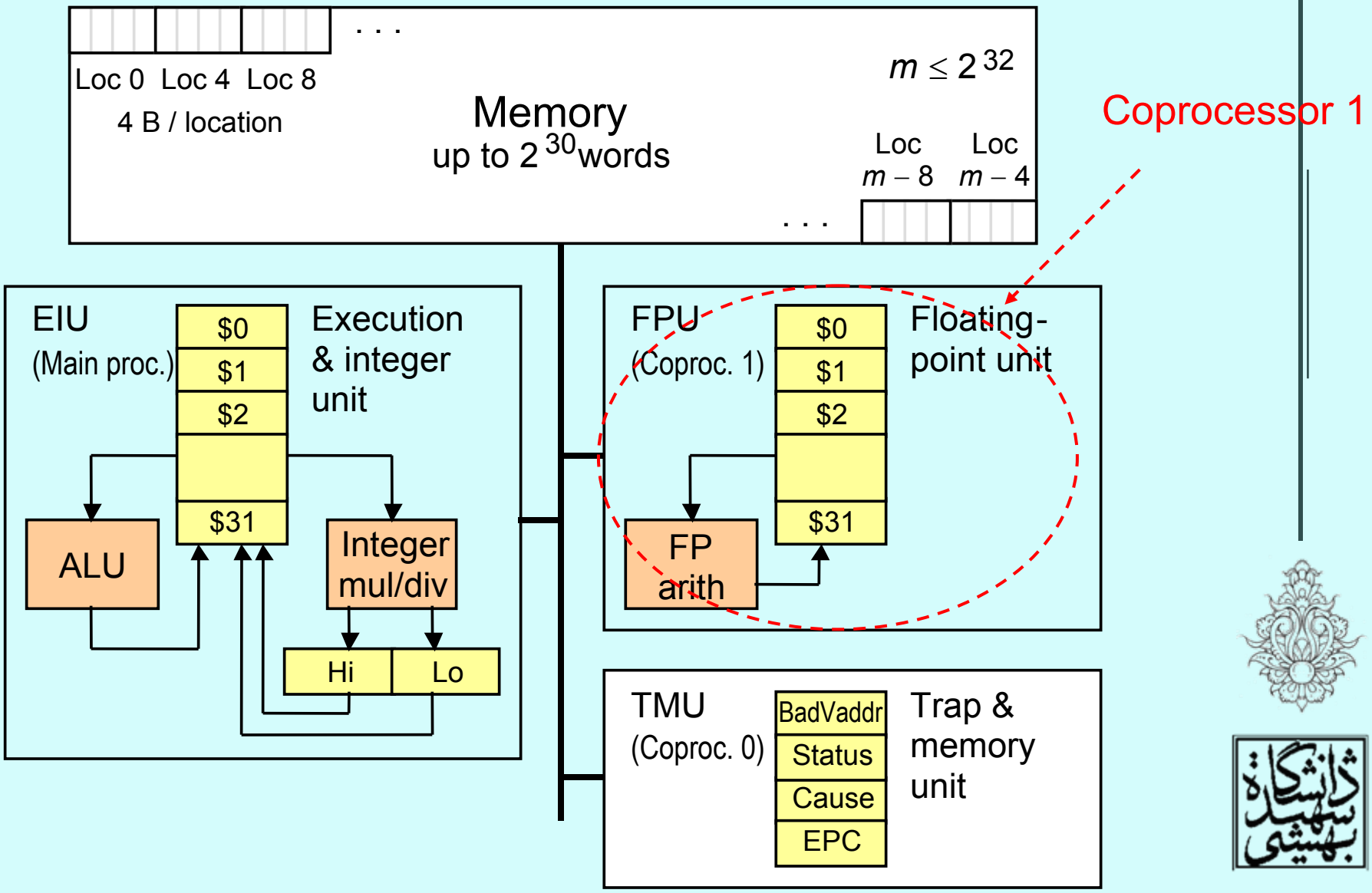
ضرب ممیز شناور

1. توان‌ها با هم جمع می‌شود
2. significandها در هم ضرب می‌شوند.
3. اعداد بهنجار شده و بروز سرریز یا فروریز چک می‌شود.
4. اعداد گرد می‌شوند و در صورت نیاز مجدد بهنجار می‌شوند.
5. علامت عدد تعیین می‌شود.



واحد ممیز شناور

- واحد محاسبات ممیز شناور معمولا اعمال جمع، تفریق، ضرب، تقسیم، معکوس‌سازی و تبدیل به صحیح را انجام می‌دهد.
- عملیات ممیز شناور به چند سیکل برای اجرا نیاز دارد.
- به صورت خط لوله نیز قابل استفاده می‌باشد.


Loc 0 Loc 4 Loc 8
4 B / location
. . .
Memory
up to 2 30 words
m ≤ 2 32
Loc m – 8
Loc m – 4
. . .
Coprocessor 1
EIU
(Main proc.)
$0
$1
$2
$31
Execution & integer unit
ALU
Integer mul/div
Hi
Lo
FPU
(Coproc. 1)
$0
$1
$2
$31
Floating-point unit
FP arith
TMU
(Coproc. 0)
BadVaddr
Status
Cause
EPC
Trap & memory unit

دستورالعمل‌های ممیز شناور در MIPS

- سی و دو ثبات جداگانه برای عملیات ممیز شناور وجود دارد:
  - $f0, $f1, … $f31
  - در صورت استفاده از دقت مضاعف این ثبات‌ها به صورت دو تایی مورد استفاده قرار می‌گیرند:
    - $f0/$f1, $f2/$f3
  - نسخه‌ی ۲ MIPS، سی‌ودو ثبات شصت و چهار بیتی دارد.
- دستورات ممیز شناور تنها بر روی ثبات‌های ممیز شناور عمل می‌کنند.

دستورالعمل‌های ممیز شناور در MIPS (ادامه...)

- دستورات خواندن و نوشتن
  - lwc1, ldc1, swc1, sdc1
    - ldc1 $f8, 32($sp)
- محاسبات با دقت معمولی
- add.s, sub.s, mul.s, div.s
  - add.s $f0, $f1, $f6
- محاسبات با دقت مضاعف
- add.d, sub.d, mul.d, div.d
  - e.g., mul.d $f4, $f4, $f6

دستورالعمل‌های ممیز شناور در MIPS (ادامه...)

- دستورات مقایسه

– `c.xx.s`, `c.xx.d` (*xx* is `eq`, `lt`, `le`, ...)
– Sets or clears FP condition-code bit
  - e.g. `c.lt.s $f3, $f4`

- دستورات پرش

– `bc1t`, `bc1f`
  - e.g., `bc1t TargetLabel`

خلاصه‌ای از دستورات ممیز شناور

	Instruction	Usage
Copy	Move s/d registers	`mov.* fd,fs`
	Move fm coprocessor 1	`mfc1 rt,rd`
	Move to coprocessor 1	`mtc1 rd,rt`
Arithmetic	Add single/double	`add.* fd,fs,ft`
	Subtract single/double	`sub.* fd,fs,ft`
	Multiply single/double	`mul.* fd,fs,ft`
	Divide single/double	`div.* fd,fs,ft`
	Negate single/double	`neg.* fd,fs`
	Compare equal s/d	`c.eq.* fs,ft`
	Compare less s/d	`c.lt.* fs,ft`
	Compare less or eq s/d	`c.le.* fs,ft`
Conversions	Convert integer to single	`cvt.s.w fd,fs`
	Convert integer to double	`cvt.d.w fd,fs`
	Convert single to double	`cvt.d.s fd,fs`
	Convert double to single	`cvt.s.d fd,fs`
	Convert single to integer	`cvt.w.s fd,fs`
	Convert double to integer	`cvt.w.d fd,fs`
Memory access	Load word coprocessor 1	`lwc1 ft,imm(rs)`
	Store word coprocessor 1	`swc1 ft,imm(rs)`
Control transfer	Branch coproc 1 true	`bc1t L`
	Branch coproc 1 false	`bc1f L`

* s/d for single/double

مثال تبدیل فارنهایت به سلسیوس

- کد به زبان C

```
float f2c (float fahr) {
  return ((5.0/9.0)*(fahr - 32.0));
}
```

- کد C کامپایل شده

```
f2c: lwc1   $f16, const5($gp)
     lwc2   $f18, const9($gp)
     div.s  $f16, $f16, $f18
     lwc1   $f18, const32($gp)
     sub.s $f18, $f12, $f18
     mul.s $f0,  $f16, $f18
     jr     $ra
```