۳.۱ استاندارد IEEE برای حساب ممیز شناور

اولین کامپیوتر بنام Z3 در سال ۱۹۴۱ توسط کُنراد تسوزه [۱] در برلین ساخته شد. Z3 یک کامپیوتر الکترومکانیکی با دستگاه اعدادش دودویی بود که در سال ۱۹۴۳ در جریان بمباران‌های جنگ جهانی دوم نابود شد. گفته می‌شود اولین کامپیوتر کاملا الکترونیکی اِنیاک [۲] است که در سال ۱۹۴۵ در دانشگاه پنسیلوانیا ساخته شد و دارای دستگاه اعداد ده‌دهی بود.

اولین استفاده‌هایی که از کامپیوتر در دهه‌ی ۱۹۵۰ می‌شد، محاسبات عددی درکاربردهای علمی بود اما در دهه‌ی ۱۹۶۰ استفاده‌ی اصلی آن‌ها در تجارت بود و نه برای محاسبات عددی. امروزه بیشتر کاربران از کامپیوترها برای پردازش اطلاعاتی چون متن، تصویر و فیلم، فایل‌های صوتی و سایر انواع اطلاعات استفاده می‌کنند بدون آن‌که اطلاع داشته باشند که پردازش چنین اطلاعاتی نیاز به حجم بزرگی از محاسبات عددی دارد.

درابتدای عصر محاسبات، حساب ممیز شناور در هر کامپیوتر بصورت خاصِ همان کامپیوتر پیاده‌سازی شده بود . در نتیجه حاصل هر محاسبه بستگی به نوع خاص کامپیوتر در حال استفاده داشت و همچنین امکان انتقال برنامه‌های نوشته‌شده در یک کامپیوتر به کامپیوترهای دیگر بسیار محدود بود یعنی برنامه ای که بسیار خوب در یک کامپیوتر اجرا می‌شد ، می‌توانست در کامپیوتر دیگری غیر قابل‌اجرا باشد. درآن دوره هم پیاده‌سازی حساب ممیزشناور در کامپیوترها متفاوت بود یعنی مثلا تعداد بیت‌های اختصاص داده‌شده به مانتیس و نما در پیاده‌سازی‌های مختلف حساب ممیز شناور تفاوت داشت، علاوه بر اینکه مبنای دستگاه اعداد کامپیوترهای مختلف متفاوت بود. جدول زیر را ببینید.

کامپیوتر	β	p	$U=-L$
IBM 7090	2	27	2^7
$Borroughs$ 5000 $Series$	8	13	2^6
IBM 360/370	16	6	2^6
DEC 11/780 VAX	2	24	2^7
$Hewlett\ Packard$ 67	10	10	99

درسال ۱۹۸۵ در نتیجه‌ی همکاری دانشمندان علوم کامپیوتر از دانشگاه‌ها و متخصصان سخت‌افزار از صنعت، یک استاندارد برای نمایش اعداد ممیزشناور دودویی و حساب ممیزشناور آن‌ها بوجود آمد. این

[۱] Konrad Zuse

[۲] ENIAC: Electronic Numerical Integrator and Computer

استاندارد که IEEE p754 نام دارد با حمایت انجمن مهندسین برق و الکترونیک که ((آی تریپل ای)) خوانده می‌شود، ارائه شد. سرپرستی دانشمندان دانشگاهی علوم کامپیوتر به عهده‌ی ویلیام کاهان [۱] استاد دانشگاه کالیفرنیا در برکلی بود و کارخانجات کامپیوتری هم‌چون Intel، Hp، Motorola و Apple نیز درگیر ایجاد این استاندارد بودند.

شکل ۶.۱: ویلیام کاهان (عکس از سایت دانشگاه کالیفرنیا در برکلی)

این استاندارد تاثیر شگرفی بر صنعت کامپیوتر داشته و ویلیام کاهان در سال ۱۹۸۹ بخاطر تلاشهایش در سرپرستی استاندارد جایزه‌ی معتبر تورینگ را که از انجمن ماشین‌آلات محاسباتی [۲] دریافت کرد. در سال ۱۹۸۵، استاندارد دیگری بنام IEEE p854 برای هر دو مبنای ۲ و ۱۰ ابداع شد.

استاندارد IEEE دو نوع پایه‌ای از اعداد ممیزشناور را معین می‌کند: یگانه [۳] و دوگانه [۴]. (نوع دیگری بنام چهارگانه [۵] نیز در استاندارد معرفی شده که در مورد آنها بحث نخواهیم کرد). در استاندارد ، تعدادی نماد خاص معرفی شده‌اند. برخی از آن‌ها عبارتنداز `NaN`[۶]، `-Inf` و `+Inf`. دو نماد `-Inf` و `+Inf` مقادیر مشابه با مفاهیم ریاضی $\pm\infty$ بوده و نماد `NaN` برای مواقعی هم‌چون `Inf - Inf` که حاصل یک عمل ممیزشناور تعریف نشده باشد، پیش‌بینی شده است. کمیت دیگری که بصورت خاص در استانداردمشخص می شود، عدد صفر است. استاندارد IEEE چگونگی انجام اعمال حساب ممیزشناور در سبک‌های مختلف گردکردن را نیز مشخص می‌کند.

ساختار کلی هر دو نوع پایه‌ای یگانه و دوگانه در مبنای دو یکی بوده و تنها در تعداد بیت‌های

[۱] William Kahan
[۲] Association of Computing Machinery (ACM)
[۳] single
[۴] double
[۵] quad
[۶] Not a Number

اختصاص‌یافته به مانتیس و همچنین تعداد بیت‌های اختصاص‌یافته به توان متفاوتند. در قالب یگانه کلا ۳۲ بیت داریم که اولی به علامت عدد [۱]، هشت بیت بعدی به توان و نهایتا ۲۳ بیت پایانی به مانتیس اختصاص یافته‌اند. پس به کمک ایده بیت پنهان در این قالب، دقت برابر است با $p = 23 + 1$ و در نتیجه اپسیلون ماشین برابر است با

$$\varepsilon_M = 2^{-(p-1)} = 2^{-23} \approx 1.2 \times 10^{-7}.$$

استاندارد مملو از جزییات دیگری است که به آن‌ها نپرداخته‌ایم از جمله اینکه به منظور ذخیره‌ی توان اعداد نرمال در هر دو قالب یگانه و دوگانه از ایده‌ای به نام توان اریب استفاده می‌شود تا جلوی اختصاص یکی از بیت‌های توان به علامت توان گرفته شود. محدوده‌ی توان دودویی اعداد نرمال در قالب یگانه بین -126 و $+127$ است. کوچک‌ترین عدد نرمال مثبت در این قالب عبارت است از

$$N_{\min} = (1.00\cdots0)_2 \times 2^{-126} = 2^{-126} \approx 1.2 \times 10^{-38}$$

و بزرگترین عدد نرمال مثبت در قالب یگانه برابر است با

$$N_{\max} = (1.11\cdots1)_2 \times 2^{+127} \approx 1.7 \times 10^{+38}.$$

به قالب دوگانه (که پیش‌فرض در بسیاری از نرم افزارهای محاسبات علمی همچون متلب است) دو برابرِ کل قالب یگانه یعنی ۶۴ بیت اختصاص داده‌شده که اولی برای ذخیره‌سازی علامت عدد، ۱۱ بیت بعدی برای توان و نهایتا ۵۲ بیت پایانی ویژه‌ی مانتیس هستند. با توجه به ایده‌ی بیت پنهان داریم $p = 52 + 1$ و در نتیجه

$$\varepsilon_M = 2^{-(p-1)} = 2^{-52} \approx 2.2 \times 10^{-16}.$$

محدوده‌ی توان دودویی اعداد نرمال در قالب دوگانه بین -1022 و $+1023$ بوده و کوچک‌ترین و بزرگ‌ترین اعداد نرمال مثبت عبارتند از:

$$N_{\min} = (1.00\cdots0)_2 \times 2^{-1022} = 2^{-1022} \approx 2.2 \times 10^{-308},$$

$$N_{\max} = (1.11\cdots1)_2 \times 2^{+1023} \approx 1.8 \times 10^{+308}.$$

[۱] sign bit

دستورهای `eps`، `realmin` و `realmax` در نرم افزار متلب به ترتیب با ε_M، $N_{\min}$ و $N_{\max}$ در نمادهای بالا (به طور پیش‌فرض در قالب دوگانه) متناظر هستند[۱].

همانگونه که دیدیم هر چه از صفر دورتر می‌شویم فاصله‌ی بین اعداد ممیزشناور نیز بیشتر می‌شود. در مورد استاندارد IEEE می‌توان این موضوع را در متلب به کمک دستور `eps(x)` که همان `ulp(x)` است و فاصله‌ی بین عدد x و نزدیک‌ترین عدد مجاور x در دستگاه اعداد ممیزشناور را مشخص می‌کند مشاهده کرد:

```
>> eps
ans =
   2.2205e-16
>> eps(1)
ans =
   2.2205e-16
>> eps(10)
ans =
   1.7764e-15
>> eps(100000)
ans =
   1.4552e-11
>> eps(realmin)
ans =
  4.9407e-324
>> eps(realmax)
ans =
  1.9959e+292
```

[۱] دقت کنید که `realmin` در قالب دوگانه کوچک‌ترین عدد ((نرمال)) مثبت است و نه کوچک‌ترین عدد مثبت. مقدار کوچک‌ترین عدد مثبت در قالب دوگانه تقریبا برابر است با 10^{-324}. این عدد، زیرنرمال می‌باشد.

این تفاوت‌ها بسیار چشم‌گیر است . با این حال فاصله‌ی نسبی اعداد ممیزشناور در سرتاسر خط اعداد ماشین تقریبا یکنواخت است:

```
>> eps(realmin)/realmin
ans =
   2.2205e-16
>> x = 10000; eps(x)/x
ans =
   1.8190e-16
>> eps(realmax)/realmax
ans =
   1.1103e-16
```

می‌توان دید که فاصله‌ی نسبی بین اعداد ماشین تقریبا برابراست با اپسیلون ماشین.

۱.۳.۱ حساب ممیز شناور

مجموعه‌ی اعداد حقیقی تحت هرچهار عمل اصلی بسته است اما یکی از مهم‌ترین مشکلات حساب ممیز شناور بسته نبودن مجموعه‌ی اعداد ماشین تحت عملیات حسابی است. یعنی اگر $x, y \in F_{\beta,p}^{L,U}$ و $* \in \{+, -, \times, /\}$ آنگاه ممکن است $x * y \notin F_{\beta,p}^{L,U}$. بعنوان مثال تقسیم یک بر سه را در نظر بگیرید که حاصل آن، در مبنای دو یا ده، دارای بینهایت رقم بوده و در نتیجه نمایش آن در دستگاه‌هایی با دقت متناهی هم‌چون $F_{2,p}^{L,U}$ یا $F_{10,p}^{L,U}$ قطعا منجر به مقداری خطا خواهد بود .[1]

پس راهی که برای انجام محاسبات روی مجموعه‌ی اعداد ماشین به ذهن می‌رسد، این است که جوابِ عمل ممیزشناور را که به طور کلی عددی در $\mathbb{R}$ است، گردکرده و با عددی متعلق به $F_{\beta,p}^{L,U}$ تقریب بزنیم. فرض کنید $* \in \{+, -, \times, /\}$ یکی از چهار عمل اصلی ریاضی بوده و $\circledast \in \{\oplus, \ominus, \otimes, \oslash\}$ عمل متناظر با $*$ در $F_{\beta,p}^{L,U}$ باشد. خوشبختانه استاندارد آی-تریپل-ای برای حساب ممیزشناور تضمین زیر را که به خاصیت

[1] چنانچه دستگاه اعدادی با مبنای سه داشتیم، جواب را می‌شد بدون خطا نمایش داد! چراکه $\frac{1}{3} = (0.1)_3 \times 3^0$.

بیشترین کیفیت معروف است برای چهار عمل اصلی ارائه می‌کند:

$$x, y \in F_{\beta,p}^{L,U}, \quad * \in \{+, -, \times, /\} \implies x \circledast y = fl(x * y).$$

به بیان دیگر حاصل عمل ممیز شناور با دقت متناهی $x \circledast y$ با سناریویی که ابتدا $x * y$ به صورت ریاضی با دقت بینهایت محاسبه شده و پاسخِ کاملا درستِ ریاضی آن که عددی در $\mathbb{R}$ است تنها یک‌بار به عدد ممیزشناورِ همسایه گرد شود مطابقت خواهد داشت.

پس تنها خطایی که در هر تک مرحله از اجرای یکی از چهارعمل اصلی در حساب ممیز شناور با دقت متناهی وجود خواهد داشت همان خطای پایانی گردکردن در $F_{\beta,p}^{L,U}$ می‌باشد. نتیجه‌ی مهم این‌که لزومی ندارد جزییات الگوریتم‌های پیاده‌سازی عملیات ممیزشناور پایه‌ای در مبنای دو را (که بیشتر مبحثی علوم کامپیوتری است تا ریاضی) بدانیم تا بتوانیم در مورد میزان درستیِ جوابی که از آنها می‌گیریم مطلع شویم! قضیه‌ی زیر نتیجه‌ی مستقیم قضیه‌ی ۱.۲.۱ و خاصیت بیشترین کیفیت حساب ممیز شناور است.

قضیه ۱.۳.۱. فرض کنید x و y دو عدد ممیزشناور نرمال باشند. در این‌صورت میزان خطای نسبی چهار عمل اصلی ممیزشناور به یکی از دو صورت زیر کران‌دار می‌شود:

- اگر از یکی از دو سبک گردکردن به پایین یا بالا استفاده شود آنگاه

$$\frac{|x * y - x \circledast y|}{|x * y|} < \varepsilon_M.$$

- اگر از سبک گردکردن به نزدیک‌ترین استفاده شود آنگاه

$$\frac{|x * y - x \circledast y|}{|x * y|} \leq \frac{\varepsilon_M}{2}.$$

□

تاکنون دو قضیه‌ی مهم را آموخته‌ایم که به ترتیب حداکثر میزان خطای گردکردن در نمایش یک عدد حقیقی و میزان خطای گردکردن در هر دفعه اجرای یکی از چهار عمل اصلی در محدوده‌ی نرمال را مشخص

می‌کرد. در مورد اول دیدیم که

$$\frac{|fl(x)-x|}{|x|} \leq u \tag{۶.۱}$$

که در سبک گردکردن به نزدیکترین $u=\epsilon_M/2$ و در سبک‌های گردکردن به پایین و بالا $u=\epsilon_M$ می‌باشد. اکنون قرار دهید

$$\delta := \frac{fl(x)-x}{x}. \tag{۷.۱}$$

پس δ می‌تواند کمیتی مثبت، منفی یا صفر باشد و طبق (۶.۱) داریم: $|\delta| \leq u$. از رابطه‌ی (۷.۱) داریم:

$$fl(x) = x\delta + x = x(1+\delta).$$

پس رابطه‌ی (۶.۱) را می‌توان به صورت معادل زیر نیز بیان کرد:

$$\boxed{fl(x) = x(1+\delta), \quad |\delta| \leq u.}$$

از سوی دیگر فرض کنید x و y دو عدد ماشین عضو دستگاه ممیزشناور $F_{\beta,p}^{L,U}$ باشند، $* \in \{+,-,\times,/\}$ یکی از چهار عمل اصلی ریاضی بوده و $\circledast \in \{\oplus,\ominus,\otimes,\oslash\}$ عمل متناظر با $*$ در $F_{\beta,p}^{L,U}$ باشند. با توجه به این‌که طبق خاصیت بیشترین کیفیت (که بعنوان نمونه در استاندارد IEEE تضمین شده) داریم

$$x \circledast y = fl(x * y)$$

و طبق قضیه‌ی قبل هم دیدیم که

$$\frac{|x*y - x\circledast y|}{|x*y|} \leq u.$$

پس برای هر دو عدد ماشین x و y به طرز مشابه با قبل داریم:

$$\boxed{x \circledast y = (x*y)(1+\delta), \quad |\delta| \leq u.}$$

به بیان سردستی، فرمول‌های بالا چیزی نیستند جز بازنویسی رابطه‌ی

$$\text{خطای نسبی} = \frac{\text{خطای مطلق}}{\text{مقدار درست}} = \frac{\text{مقدار درست} - \text{مقدار تقریبی}}{\text{مقدار درست}}$$

به صورت

$$\text{مقدار تقریبی} = (\text{مقدار درست}) \times (1 + \text{خطای نسبی}).$$