# مبانی برنامه نویسی

## نمایش اعداد در کامپیوتر

Dr. Mehran Safayani

safayani@iut.ac.ir

safayani.iut.ac.ir

https://www.aparat.com/mehran.safayani

https://github.com/safayani/Programming_Basics_course

Department of Electrical and computer engineering,  Isfahan university of technology, Isfahan, Iran

$$2586_{10} = (2 * 10^3) + (5 * 10^2) + (8 * 10^1) + (6 * 10^0)$$

# سیستم اعداد دهدهی

سیستم اعداد دهدهی یکی از سیستم‌های اعداد متداول است که همگان روزانه با آن سروکار دارند.
در این سیستم هر عدد می‌تواند ترکیبی از ارقام ۰ تا ۹ باشد. هر عدد در سیستم دهدهی را بنابر آنچه قبلا نشان داده شد
می‌توان به صورت زیر نمایش داد:

$$A = a_{n-1} \ldots a_1 a_0 a_{-1} a_{-2} \ldots a_{-m} = \sum_{k=-m}^{n-1} a_k . (10)^k$$

- مثال

$$123.45 = (1 * 10^2) + (2 * 10^1) + (3 * 10^0) + (4 * 10^{-1}) + (5 * 10^{-2})$$

# سیستم اعداد دودویی

در این سیستم، مبنای اعداد، ۲ است. لذا هر عدد در این سیستم می‌تواند ترکیبی از ارقام ۰ و ۱ باشد. مانند ۱۱۱، ۱۰۱۱ و ۱۰۰۰۱۱۱.

شکل کلی اعداد این سیستم به صورت زیر می‌باشد.

$$\sum_{k=-m}^{n-1} a_k \, . \, (2)^k$$

# مثال‌هایی از تبدیل اعداد دودویی به دهدهی

$$(11001)_2 = (1 * 2^4) + (1 * 2^3) + (0 * 2^2) + (0 * 2^1) + (1 * 2^0)$$

$$= 16 + 8 + 0 + 0 + 1 = (25)_{10}$$

---

$$(1110.01)_2 = (1110.0)_2 + (0.01)_2$$

$$(1110.0)_2 = (1 * 2^3) + (1 * 2^2) + (1 * 2^1) + (0 * 2^0) = 8 + 4 + 2 + 0 = (14)_{10}$$

$$(0.01)_2 = (0 * 2^{-1}) + (1 * 2^{-2}) = 0 + \frac{1}{4} = (0.25)_{10}$$

$$\longrightarrow (1110.01)_2 = (14)_{10} + (0.25)_{10} = (14.25)_{10}$$

# سیستم اعداد هشتایی (octal)

در این سیستم، مبنای اعداد، ۸ است. لذا هر عدد در این سیستم می‌تواند ترکیبی از ارقام ۰ تا ۷ باشد.
نمایش عدد در این سیستم به این صورت است.

$$\sum_{k=-m}^{n-1} a_k \,.\, (8)^k$$

---

- مثال

$$(2057)_8 = (2 * 8^3) + (0 * 8^3) + (5 * 8^1) + (7 * 8^0)$$

$$= 1024 + 0 + 40 + 7 = (1071)_{10}$$

# سیستم اعداد هشتایی (octal)

- مثال

$$(4706)_8 = (?)_{10}$$

Common values multiplied by the corresponding digits

$$(4706)_8 = (4 * 8^3) + (7 * 8^2) + (0 * 8^1) + (6 * 8^0)$$

$$= (4 * 512) + (7 * 64) + (0) + (6 * 1)$$

$$= 2048 + 448 + 0 + 6$$

**Sum of these products**

$$= (2502)_{10}$$

# سیستم اعداد شانزده تایی (هگزادسیمال)

در این سیستم، مبنای اعداد، ۱۶ است. لذا می‌توان از ۱۶ رقم در نوشتن اعداد این سیستم استفاده کرد. چون اعداد ۹ به بالا را به عنوان یک رقم نمی‌شناسیم، برای نمایش ۱۰ تا ۱۵ از علائم A تا F استفاده می‌کنیم. لذا ارقام مبنای ۱۶ شامل ۱ تا ۹ و حروف A تا F می‌شود. اعدادی مثل ACE، 1CD و B1A اعدادی در مبنای ۱۶ هستند. شکل کلی نمایش این اعداد به صورت

$$\sum_{k=-m}^{n-1} a_k \cdot (16)^k$$

- مثال

$$(1AF)_{16} = (1 * 16^2) + (A * 16^1) + (F * 16^0) = (1 * 256) + (10 * 16) + (15 * 1)$$

$$= 256 + 160 + 15 = (431)_{10}$$

# سیستم اعداد شانزده تایی (هگزادسیمال)

| Hexadecimal | Decimal | Binary |
|---|---|---|
| A | 10 | 1010 |
| B | 11 | 1011 |
| C | 12 | 1100 |
| D | 13 | 1101 |
| E | 14 | 1110 |
| F | 15 | 1111 |

# سیستم اعداد

در سیستم‌های عدد معمولی، موقعیت مکانی هر رقم دارای ارزش معینی است. در چنین سیستم‌هایی می‌توان هر عدد را به صورت زیر نمایش داد:

$$N = (a_{n-1} \ \dots a_1 a_0 a_{-1} a_{-2} \ \dots a_{-m})_B$$

$$N = a_{n-1}B^{n-1} + \cdots + a_0B^0 + a_{-1}B^{-1} + a_{-2}B^{-2} + \cdots + a_{-m}B^{-m}$$

$$N = \sum_{k=-m}^{n-1} a_k B^k$$

در نمایش فوق داریم:

B : مبنای سیستم است و $0 \le a_k \le B - 1$

m : تعداد ارقام اعشاری

n : تعداد ارقام صحیح

ضرایب : $a_0, a_1, a_2, \ \dots$

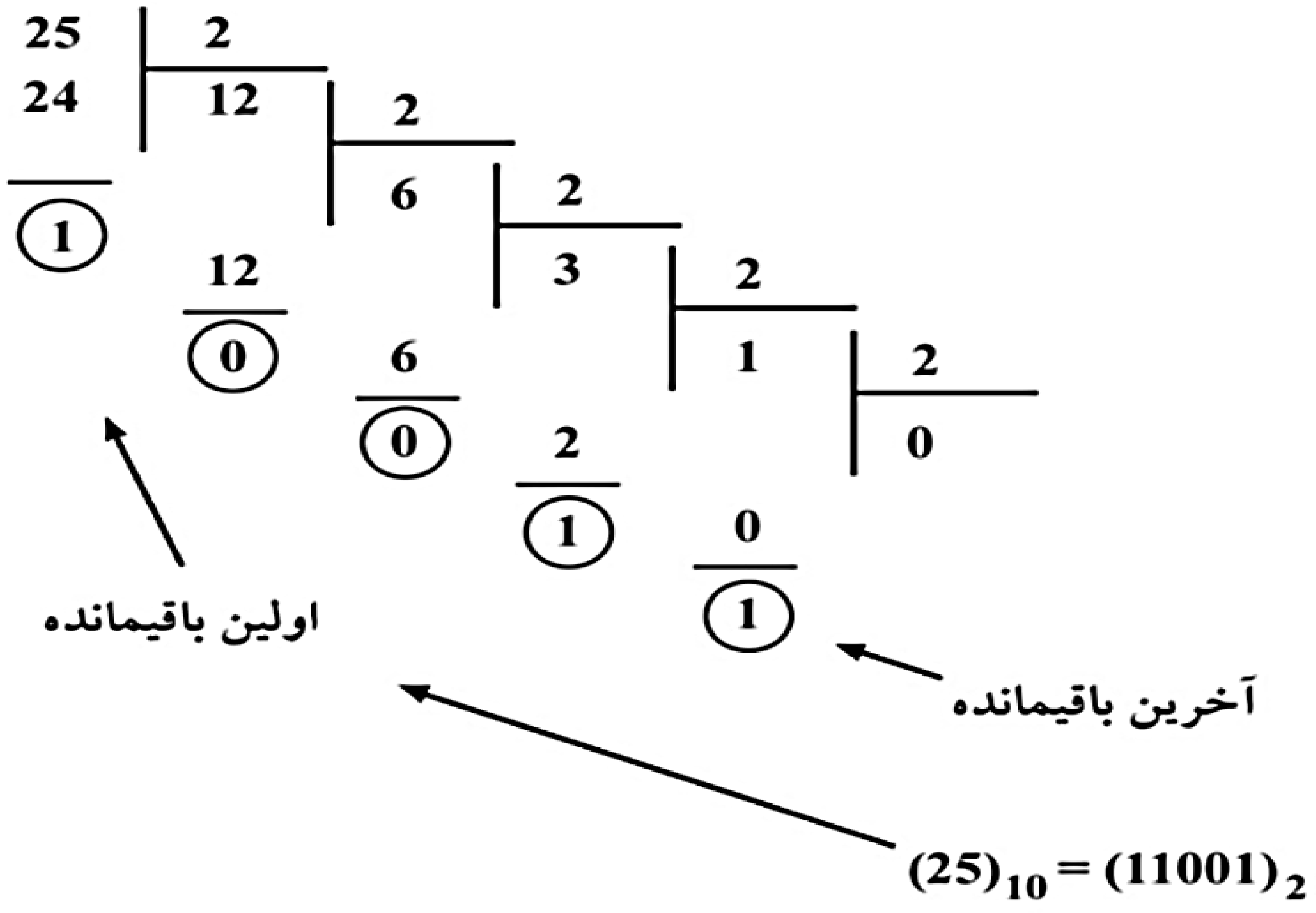
کمترین مقداری که B می‌تواند اختیار کند، برابر ۲ است.

# مثال‌هایی از سیستم اعداد

- مبنای ۱۰ به ۲

$$(25)_{10} = (?)_2$$

جواب:



# مثال‌هایی از سیستم اعداد

- مبنای ۱۰ به ۸

$$(\mathbf{952})_{\mathbf{10}} = (?)_{\mathbf{8}}$$

جواب:

| 8 | 952 | باقی‌مانده |
|---|---|---|
| | 119 | 0 |
| | 14 | 7 |
| | 1 | 6 |
| | 0 | 1 |

$$(952)_{10} = (1670)_8$$

# مثال‌هایی از سیستم اعداد

- مبنای ۶ به ۴

$$(\mathbf{545})_{\mathbf{6}} = (?)_{\mathbf{4}}$$

جواب:

مرحله اول تبدیل از مبنای ۶ به ۱۰

$$(545)_6 = (5 * 6^2) + (4 * 6^1) + (5 * 6^0)$$

$$= (5 * 36) + (4 * 6) + (5 * 1)$$

$$= 180 + 24 + 5$$

$$= (209)_{10}$$

مرحله دوم تبدیل از مبنای ۱۰ به ۴

| 4 | 209 | باقی‌مانده |
|---|---|---|
| | 52 | 1 |
| | 13 | 0 |
| | 3 | 1 |
| | 0 | 3 |

$$(209)_{10} = (3101)_4$$

$$(545)_6 = (3101)_4$$

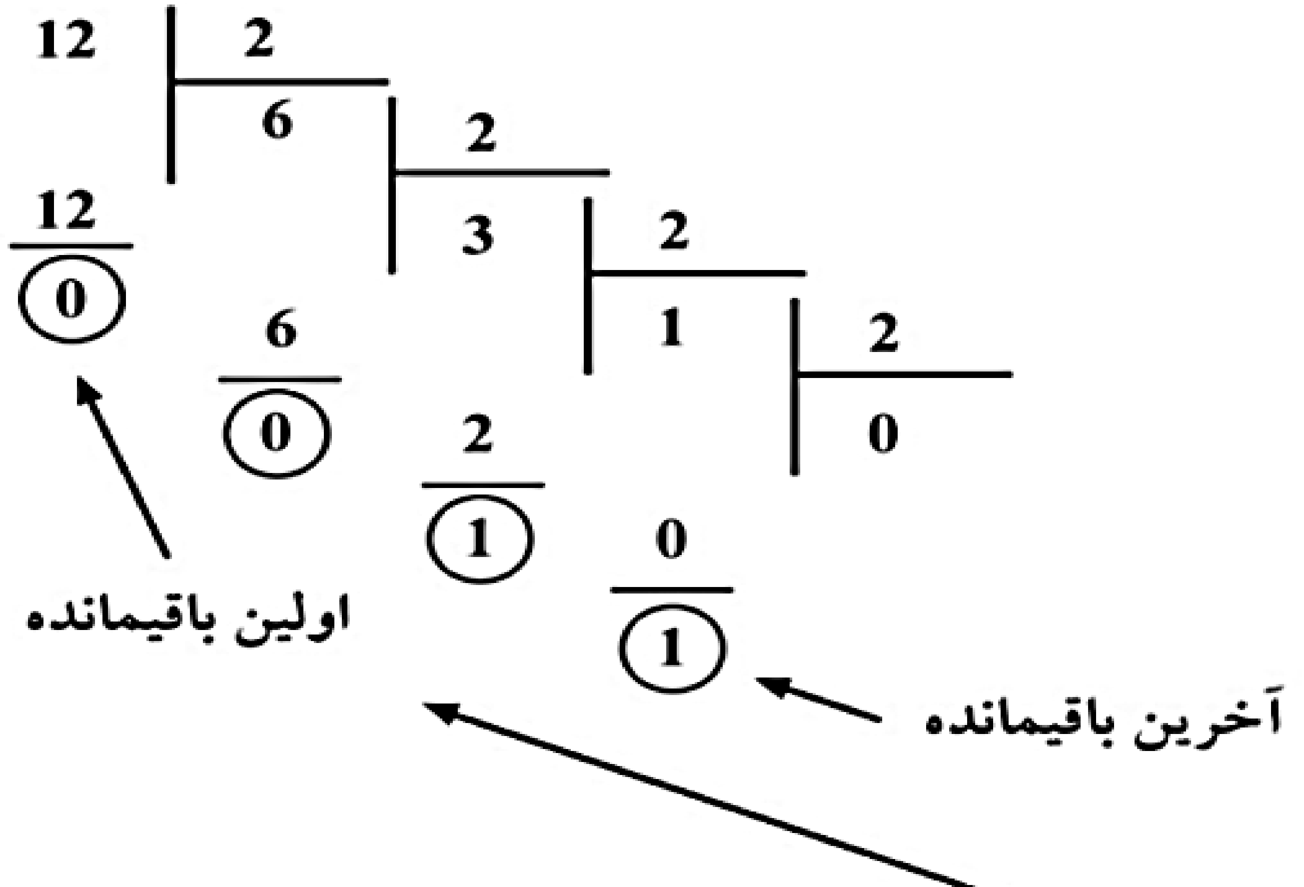
# مثال‌هایی از سیستم اعداد

- مبنای ۱۰ به ۲ اعداد اعشاری

$$(12.25)_{10} = (?)_2$$

جواب:

مرحله اول بدست آوردن مبنا ۲ قسمت صحیح عدد




$(12)_{10} = (1100)_2$

# مثال‌هایی از سیستم اعداد

مرحله دوم بدست آوردن مبنا ۲ قسمت اعشار عدد: با استفاده از ضرب‌های متوالی قسمت اعشار عدد را به عدد صحیح تبدیل می‌کنیم

1) $0.25 * 2 = 0.50 \rightarrow 0.50 * 2 = 1.00 \rightarrow (0.01)_2$

با جمع دو مقادیر بدست آمده از دو مرحله انجام شده، مبنای ۲ عدد ۱۲.۲۵ برابر با ۱۱۰۰.۰۱ می‌شود.

2) $0.75 * 2 = 1.50 \rightarrow 0.50 * 2 = 1.00 \rightarrow (0.11)_2$

3) $0.3 * 2 = 0.60 \rightarrow 0.60 * 2 = 1.20 \rightarrow 0.2 * 2 = 0.4 \rightarrow 0.4 * 2 = 0.8 \rightarrow 0.8 * 2 = 1.6$

$\rightarrow 0.6 * 2 = 1.2 \; \ldots \rightarrow (0.0\overline{1001})_2$

نکته: اگر با ضرب‌های متوالی، قسمت اعشار به صفر نرسد، باید عمل ضرب را تا پر شدن کلمه‌ی حافظه ادامه داد. **اگر عدد اعشار دارای تناوب شد از یک خط تیره بالای عدد برای نشان دادن تناوب استفاده می کنیم.**

برای تبدیل اعداد اعشاری مبنای ۲ به مبنای ۱۰ از بسط عدد استفاده می‌کنیم.

## مثال‌هایی از سیستم اعداد

- مبنای ۲ به ۸

$$(\mathbf{11001})_{\mathbf{2}} = (?)_{\mathbf{8}}$$

جواب:

صفر اضافه شده $\longrightarrow$ $(\underbrace{011}_{3}\ \ \underbrace{001}_{1})_2 = (31)_8$

$$(\mathbf{10011.1101})_{\mathbf{2}} = (?)_{\mathbf{8}}$$

جواب:

صفر اضافه شده $\longrightarrow$ $(\underbrace{010}\ \ \underbrace{011}\,.\,\underbrace{110}\ \ \underbrace{100})_2 = (23.64)_8$

صفرای اضافه شده به قسمت اعشار

## مثال‌هایی از سیستم اعداد

- مبنای ۸ به ۲

$$(\mathbf{25.34})_{\mathbf{8}} = (?)_{\mathbf{2}}$$

جواب:

$$(25.34)_8 = (010101.011100)_2 = (10101.0111)_2$$

- مبنای ۲ به ۱۶

$$(\mathbf{01111101.0110})_{\mathbf{2}} = (?)_{\mathbf{16}} \xrightarrow{\text{جواب}} (01111101.0110)_2 = (7D.6)_{16}$$

- مبنای ۱۶ به ۲

$$(\boldsymbol{F}\mathbf{25.03})_{\mathbf{16}} = (?)_{\mathbf{2}} \xrightarrow{\text{جواب}} (F25.03)_{16} = (111100100101.00000011)_2$$

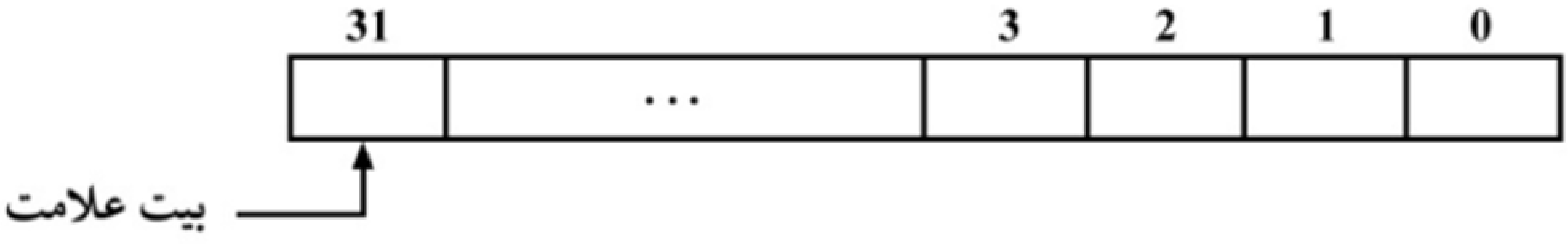
# انجام چند عمل جمع در مبنای  و ۱۶

$$\begin{array}{r} 100111+ \\ 111010 \\ \hline 1100001 \end{array} \qquad \begin{array}{r} 1111011+ \\ 1010000 \\ \hline 11001011 \end{array} \qquad \begin{array}{r} 11111+ \\ 11110 \\ \hline 111101 \end{array}$$

---

$$\begin{array}{r} 1A53\ + \\ 371 \\ \hline 1DC4 \end{array} \qquad \begin{array}{r} ABE12\ + \\ 354 \\ \hline AC166 \end{array}$$

# نگهداری اعداد علامت‌دار در کامپیوتر




نمایش عدد صحیح

# نگهداری اعداد صحیح منفی

برای نمایش اعداد صحیح منفی می‌توان به سه روش عمل کرد:

۱. روش علامت و مقدار
۲. روش متمم ۱
۳. روش متمم ۲

# روش علامت مقدار

در این روش، اعداد منفی مانند اعداد مثبت ذخیره می‌شوند؛ با این تفاوت که در بیت علامت مقدار یک قرار می‌گیرد.  در زیر مثالی از نمایش عدد ۲۱- به طول کلمات ده بیت آمده است:

| 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |

# روش علامت مقدار

این روش دارای دو اشکال عمده است:

1. برای صفر منفی و صفر مثبت دو نمایش جداگانه وجود دارد.

| 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 |

نمایش صفر منفی

| 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |

نمایش صفر مثبت

۲. برای عمل تفریق باید مدار جداگانه‌ای طراحی شود.

نکته: اگر طول هر کلمه ماشین M فرض شود، بزرگ‌ترین و کوچک‌ترین اعداد قابل نمایش به روش علامت و مقدار به صورت زیر است:

کوچکترین: $-(2^{M-1}-1)$ بزرگترین: $2^{M-1}-1$

پس بازه اعداد قابل نمایش را می توان $[-(2^{M-1}-1), 2^{M-1}-1]$ در نظر گرفت

# روش متمم ۱

برای نمایش اعداد منفی به روش متمم ۱، عدد را به مبنای ۲ تبدیل کرده، نمایش مثبت عدد را مشخص می‌کنیم و در نمایش حاصل، تمام ۰ ها را ۱ و تمام ۱ ها را به ۰ تبدیل می‌کنیم و یا به عبارت دیگر تمام ارقام را از ۱ کم می کنیم (۲ مبنای عدد است). به عنوان مثال نمایش عدد ۱۹- بصورت زیر است:

$$-19 = (-10011)_2$$

| 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 0 |

نمایش عدد ۱۹ – به روش متمم ۱

| 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 |

نمایش عدد ۱۹

در این حالت نیز برای نمایش صفر مثبت و صفر منفی دو نمایش مختلف وجود دارد ولی برای انجام عمل تفریق نیاز به مدار جداگانه‌ای نیست، یعنی روش متمم ۱، اشکال دومِ روش علامت و مقدار را برطرف می‌کند

پس بازه اعداد قابل نمایش را می توان $[-(2^{M-1}-1), 2^{M-1} - 1]$ در نظر گرفت

# روش متمم ۲

این روش، هر دو اشکال روش علامت و مقدار را حل می‌کند. در این روش باید به صورت عمل کرد:

1. نمایش مثبت عدد
2. پیدا کردن متمم ۱ عدد
3. افزودن یک واحد به عدد حاصل

به عنوان مثل نمایش اعداد ۲۳- در ماشینی به طول کلمات ده بیت بدین صورت است: $-23 = (-10111)_2$

| 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 1 |

نمایش عدد ۲۳

| 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |

نمایش عدد ۲۳–به روش متمم ۱

**نکته:** بازه اعداد قابل نمایش را می توان $[-2^{M-1}, (2^{M-1}-1)]$ در نظر گرفت

# جمع و تفریق در مکمل ۲

$5 + (-3) = 2$

| | | | |
|---|---|---|---|
| | 0000 0101 | = | +5 |
| + | 1111 1101 | = | −3 |
| | 0000 0010 | = | +2 |

$7 - 12 = (-5)$

| | | | |
|---|---|---|---|
| | 0000 0111 | = | +7 |
| + | 1111 0100 | = | −12 |
| | 1111 1011 | = | −5 |

# اشباع (overflow)

اگر جمع دو عدد مثبت (منفی) یک عدد منفی (مثبت) شود **اشباع** رخ داده است. همچنین جمع یک عدد مثبت و منفی هیچ‌گاه **اشباع** ندارد.

- 104 + 45 = 149:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1 | 1 | | 1 | | | | |
| | 0 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| + | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |

Overflow, no carryout. Sum is not correct.

- -103 + -69 = -172:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| $\boxed{1}$ | | | 1 | 1 | 1 | | 1 | 1 | |
| | | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 |
| | + | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| | | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 |

Overflow, with incidental carryout. Sum is not correct

- 10 + -3 = 7:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $\boxed{1}$ | 1 | 1 | 1 | 1 | | | | |
| | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 |
| + | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 |
| | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 |

Carryout without overflow  Sum is carrect.

- -19 + -7 = -26:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $\boxed{1}$ | 1 | 1 | 1 | 1 | | | 1 | |
| | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
| + | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0 |

Carryout without overflow. Sum is carrect.

- 127 + 1 = 128:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | 1 | 1 | 1 | 1 | 1 | 1 | 1 | |
| | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| + | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 |
| | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Overflow, no carryout. Sum is not correct.

- -39 + 92 = 53:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $\boxed{1}$ | 1 | | 1 | 1 | | | | |
| | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| + | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 0 |
| | 0 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |

Carryout without overflow. Sum is correct.

- $44 + 45 = 89$:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 1 | | 1 | 1 | | | |
| | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 |
| + | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |

No overflow nor carryout.

$-1 + 1 = 0$

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| $\boxed{1}$ | 1 | 1 | 1 | 1 | | | 1 | |
| | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| + | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 1 |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Carryout without overflow. Sum is carrect.

- $-75 + 59 = -16$:

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | 1 | 1 | 1 | 1 | 1 | 1 | |
| | 1 | 0 | 1 | 1 | 0 | 1 | 0 | 1 |
| + | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

No overflow nor carryout.

# نمایش اعداد: فرمت IEEE 754

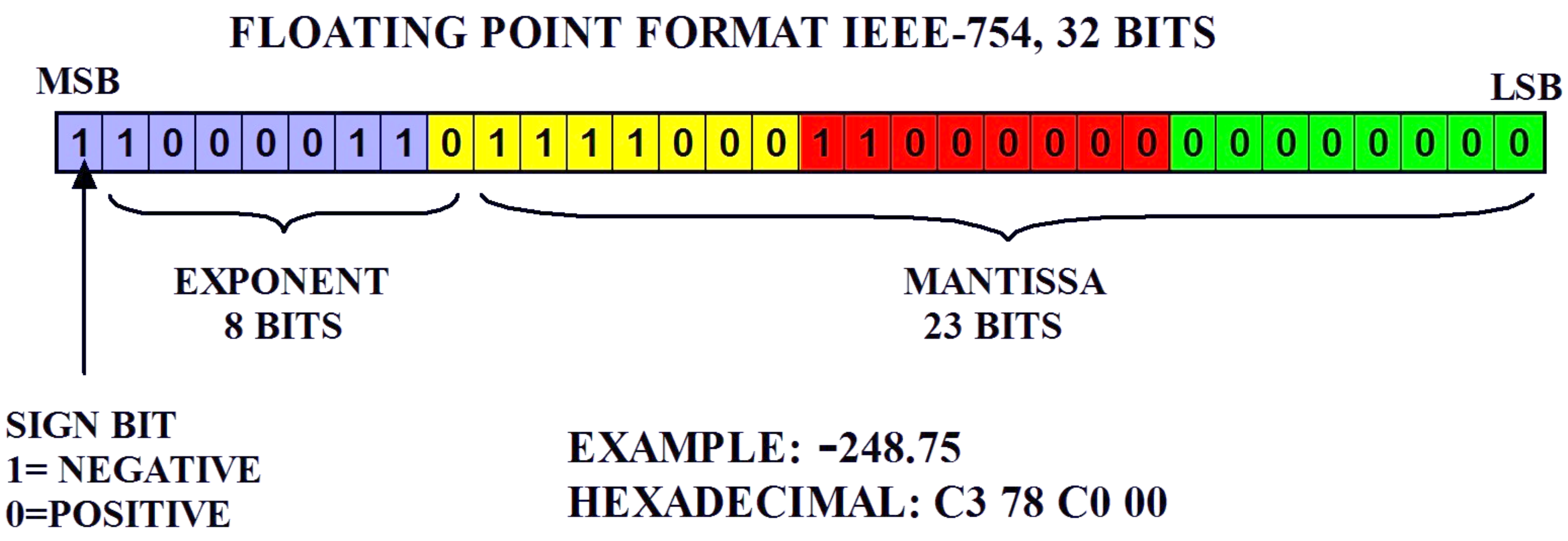
نمایش ۳۲ بیتی (Single Precision):
- علامت ۱ بیت
- نما ۸ بیت
- مانتیس ۲۳ بیت
- بایاس: ۱۲۷

نمایش ۶۴ بیتی (Double Precision) :
- علامت ۱ بیت
- نما ۱۱ بیت
- مانتیس ۵۲ بیت
- بایاس: ۱۰۲۳

| | single: 8 bits<br>double: 11 bits | single: 23 bits<br>double: 52 bits |
|---|---|---|
| S | Exponent | Fraction |

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

Single: Bias = 127; Double: Bias = 1023



# نمایش اعداد: فرمت IEEE 754

| | single: 8 bits<br>double: 11 bits | single: 23 bits<br>double: 52 bits |
|---|---|---|
| S | Exponent | Fraction |

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

- مثال: نمایش عدد با شش رقم اعشار

$$(45.45)_{10} = (?)_2$$

جواب:

| | | |
|---|---|---|
| $0.45 * 2$ | 0 | 0.9 |
| $0.90 * 2$ | 1 | 0.8 |
| $0.80 * 2$ | 1 | 0.6 |
| $0.60 * 2$ | 1 | 0.2 |
| $0.20 * 2$ | 0 | 0.4 |
| $0.40 * 2$ | 0 | 0.8 |

$$(45)_{10} = (101101)_2$$

$$(0.45)_{10} = (011100)_2$$

$$(45.45)_{10} = (101101.011100)_2$$

# نمایش ممیز شناور ۳۲ بیتی به فرمت IEEE

| | single: 8 bits<br>double: 11 bits | single: 23 bits<br>double: 52 bits |
|---|---|---|
| S | Exponent | Fraction |

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

$$(45.45)_{10}=(101101.011100)_2$$

مرحله ۱: عدد را نرمال می‌کنیم.

مرحله ۲: نما و مانتیس را بدست می آوریم

مرحله ۳: نمای بایاس شده را با اضافه کردن ۱۲۷ پیدا می‌کنیم

مرحله ۴: بیت علامت را در صورت مثبت بودن ۰ و در غیر این صورت ۱ قرار می دهیم.

# نمایش ممیز شناور ۳۲ بیتی به فرمت IEEE

| single: 8 bits<br>double: 11 bits | | single: 23 bits<br>double: 52 bits |
|---|---|---|
| S | Exponent | Fraction |

- مثال

$$(45.45)_{10} = (101101.011100)_2$$

$$(101101.011100)_2 = 1.01101011100 \ * 2^5$$

→ bias exponent = 5+127 = 132

$$\rightarrow mantissa = 01101011100$$

| | | |
|---|---|---|
| 0 | 10000100 | 01001011100000000000000 |
| Sign Bit | Biased Exponent | Trialing Significand bit or Mantissa |
| 1-bit | 8-bit | 23-bit |

4225C000

# نمایش ممیز شناور ۳۲ بیتی به فرمت IEEE

- تبدیل ۰.۸۱۲۵- به باینری بصورت single و double

| | |
|---|---|
| $0.8125 \ * 2$ | $1.625$ |
| $0.625 \ * 2$ | $1.25$ |
| $0.25 \ * 2$ | $0.5$ |
| $0.5 \ * 2$ | $1.0$ |

$$\boxed{0.825 = (0.1101)_2 = \frac{1}{2}+\frac{1}{4}+\frac{1}{16} = \frac{13}{16}}$$

$$Fraction = (0.1101)_2 = (0.1101)_2 * 2^{-1} \ (Normalized)$$

$$Exponent = -1 + Bias = 126 \ (single\ precision) \ and \ 1022 \ (double)$$

| | |
|---|---|
| 1 0 1 1 1 1 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | Single Precision |
| 1 0 1 1 1 1 1 1 1 1 1 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | Double Precision |
| 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | |

# نمایش ممیز شناور ۳۲ بیتی به فرمت IEEE

- نمایش زیر بصورت Single Precision است، این عدد در مبنای ۱۰ (decimal) چقدر است؟

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

جواب:

$\boldsymbol{Sign} = 1\ is\ negative$

$\boldsymbol{Exponent} = (01111100)_2 = 124, \boldsymbol{E - bias} = 124 - 127 = -3$

$\boldsymbol{Significand} = (1.0100 \ldots 0)_2 = 1 + 2^2 = 1.25\ (\boldsymbol{1.\ is\ implicit})$

$\boldsymbol{Value\ in\ decimal} = -1.25 * 2^{-3} = \boldsymbol{-0.15625}$

# نمایش ممیز شناور ۳۲ بیتی به فرمت IEEE

- این عدد در مبنای ۱۰ (decimal) چقدر است؟

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

جواب:

$$\textit{Value in decimal} = +(\overset{\textit{implicit}}{1}.01001100\ldots0)_2 * 2^{130-127} = (1.01001100\ldots0)_2 * 2^3$$

$$= (1010.01100\ldots0)_2 = \mathbf{10.375}$$

# نمایش ممیز شناور ۳۲ بیتی به فرمت IEEE

- نمایش زیر بصورت Double Precision است، این عدد در مبنای ۱۰ (decimal) چقدر است؟

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

جواب:

$\boldsymbol{Value\ of\ exponent} = +(10000000101)_2 - Bias = 1029 - 1023 = 6$

$\boldsymbol{Value\ of\ double\ flote} = +(1.0010101 \ldots 0)_2 * 2^6\ (1.\,is\ implicit)$

$= (1001010.10 \ldots 0)_2 = \mathbf{74.5}$

# نمایش ممیز شناور ۳۲ بیتی به فرمت IEEE

- این عدد در مبنای ۱۰ (decimal) چقدر است؟ **(سوال کلاسی)**

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**جواب:** $-1.5 * 2^{-7} = -0.01171875$

# نمایش ممیز شناور ۳۲ بیتی به فرمت IEEE

- صفر
  - فیلد توان E=0 و کسر F=0
  - +0 و -0 طبق بیت علامت S امکان پذیر است.
- بی‌نهایت
  - بی‌نهایت یک مقدار ویژه است که با حداکثر E و F=0 نمایش داده می‌شود.
  - برای Single Precision با توان ۸ بیتی حداکثر E برابر با ۲۵۵ و برای Double Precision برابر ۲۰۴۷ است.
  - بی‌نهایت می‌تواند از overflow یا تقسیم بر صفر حاصل شود. همچنین $+\infty$ و $-\infty$ طبق بیت علامت S امکان پذیر است.
- NaN (عدد نیست)
  - NaN یک مقدار ویژه است که با حداکثر E و $F \neq 0$ نمایش داده می‌شود.
  - از موقعیت‌هایی استثنایی، مانند 0/0 یا sqrt (منفی) بدست می آِید. همچنین هر عملیاتی بر روی NaN برابر با Nan است

# نمایش ممیز شناور ۳۲ بیتی به فرمت IEEE

- بزرگترین عدد اعشاری نرمال شده چیست؟

جواب برای Single Precision :

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

$\boldsymbol{Exponent\ - bias} = 254 - 127 = 127\ (largest\ exponent\ for\ SP)$

$\boldsymbol{Signifcand} = +(1.111 \dots 1)_2 = almost\ 2$ ; $\boldsymbol{Value\ in\ decimal} \approx 2 * 2^{127} \approx 2^{128} \approx \boldsymbol{3.4028 \dots * 10^{38}}$

جواب برای Double Precision :

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |

$\boldsymbol{Value\ in\ decimal} \approx 2 * 2^{1023} \approx 2^{1024} \approx \boldsymbol{1.79769 \dots * 10^{308}}$

Overflow: توان، برای قرار گرفتن در فیلد توان بسیار بزرگ است.

- کوچکترین عدد اعشاری نرمال شده (بصورت قدرمطلق) چیست؟

جواب برای Single Precision :

`0 | 00000001 | 00000000000000000000000`

$\boldsymbol{Exponent\ - bias} = 1 - 127 = -126\ (\mathit{smallest\ exponent\ for\ SP})$

$\boldsymbol{Signifcand} = +(1.000\ldots0)_2 = \mathbf{1}\ ;\ \boldsymbol{Value\ in\ decimal} \approx 1 * 2^{-126} \approx \mathbf{1.17549\ldots * 10^{-38}}$

جواب برای Double Precision :

`0 | 00000000001 | 00000000000000000000`
`00000000000000000000000000000000`

$\boldsymbol{Value\ in\ decimal} \approx 1 * 2^{-1022} \approx \mathbf{2.22507\ldots * 10^{-308}}$

Underflow: توان، برای قرار گرفتن در فیلد توان بسیار کوچک است.

# نمایش ممیز شناور ۳۲ بیتی به فرمت IEEE

کوچکترین عدد نرمال مثبت:

$$1.0 \times 2^{-126} \approx 1.175 \times 10^{-38}$$

اعداد Denormal

وقتی **exponent field = 0** و $f \neq 0$:

$$(-1)^s \times 0.f \times 2^{1-bias}$$

**بزرگترین عدد denormal مثبت:**

$$0.111...1 \times 2^{-126}$$

$$= (1 - 2^{-23}) \times 2^{-126} \approx 1.175 \times 10^{-38}$$

کوچکترین عدد denormal

$$\text{مقدار} = 0.00000000000000000000001_2 \times 2^{-126}$$

$$= 2^{-23} \times 2^{-126} = 2^{-149}$$