



КУРСОВА РАБОТА

НА ТЕМА:

**МАШИННА АРИТМЕТИКА – АРИТМЕТИЧНИ ДЕЙСТВИЯ С ФИКСИРАНА
И ПЛАВАЩА ЗАПЕТАЯ**

СТУДЕНТ: *Петя Миткова Янева*

ФАКУЛТЕТЕН НОМЕР: *121223006*

СПЕЦИАЛНОСТ: *КОМПЮТЪРНО И СОФТУЕРНО ИНЖЕНЕРСТВО*

ГРУПА: *39*

Съдържание

Въведение	3
Фиксирана запетая	3
Въведение във фиксираната запетая.....	3
Избор на брой битове.....	3
Ограничена прецизност на фиксираната запетая.....	3
Методи за пресмятане на числата.....	4
Пример за пресмятане на число.....	5
Минимални и максимални стойности	5
Предимства на фиксираната запетая	6
Недостатъци на фиксираната запетая.....	6
Плаваща запетая.....	7
Представяне на числа с плаваща запетая	7
Гъвкавост на плаващата запетая	7
Минимални и максимални стойности в плаващата запетая.....	8
Точност на плаващата запетая.....	9
Ограничения на представянето с плаваща запетая	9
Стандарт на IEEE за плаваща запетая	9
Специални стойности на IEEE 754	10
Предимства на плаваща запетая.....	10
Недостатъци на плаваща запетая.....	11
Сравнение на фиксирана и плаваща запетая	11
Заключение	12
Литература	13

1. Въведение

Машинната аритметика е основата на всяко изчисление в компютърните системи и играе ключова роля в обработката на числови данни. В съвременната компютърна наука има два основни метода за представяне на реални числа: фиксирана запетая и плаваща запетая. Всеки от тези методи има своите предимства и недостатъци, които се проявяват в зависимост от типа на изчисленията и изискванията за прецизност и точност в различни приложения.

Въпросът за избор на подходящ метод за представяне на числа е важен, тъй като той влияе на ефективността, точността и производителността на изчисленията в компютърните системи. Разбирането на принципите и прилагането на тези методи е от съществено значение за оптимизацията на изчисленията в съвременните технологии.

2. Фиксирана запетая

2.1. Въведение във фиксираната запетая

Фиксираната запетая представлява система за представяне на числа, при която позицията на десетичната запетая е фиксирана и не може да се променя. В този метод се използва определен брой бита за целочислената част и за дробната част на числото. Например, ако имаме фиксирано представяне на вида **IIII.FFFF**, това означава, че числото е разделено на две части: целочислена част (**IIII**) и дробна част (**FFFF**). Броят на знаците след запетаята е постоянен, което гарантира определена прецизност, но и налага ограничения върху диапазона на представените стойности. {3.}

2.2. Избор на брой битове

При избора на броя на битовете за целочислената и дробната част се взема под внимание не само прецизността, която искаме да постигнем, но и диапазонът на стойностите, които ще се представят. Например, ако искаме да представим по-голям диапазон стойности, ще трябва да увеличим броя на битовете за целочислената част, а ако искаме по-висока прецизност за дробната част, ще увеличим броя на битовете за дробната част. Въпреки това, увеличаването на броя на битовете води до повишаване на изчислителната сложност, както и увеличаване на паметта, необходима за съхранение на числата. {3., 5.}

2.3. Ограничена прецизност на фиксираната запетая

В този случай минималната стойност, която може да бъде представена, е 0000.0001, а максималната стойност е 9999.9999. Това обаче ограничава възможността за представяне на числа извън този диапазон и може да доведе до грешки при изчисления с числа, които не се

побират в зададения формат. Поради ограничен брой битове, могат да възникнат грешки от окръгляване. Затова е важно да се изберат подходящи стойности за броя битове, които да отговорят на конкретните изисквания на приложението или изчислението.



2.4. Методи за пресмятане на числата

Числата в представянето с фиксирана запетая могат да се изразяват по различни начини, в зависимост от метода за представяне на знака. Това са основните методи:

- **Прав код (Sign Representation):** В този метод най-старшият бит (първият бит отляво) е запазен за знака, като 0 означава положително число, а 1 - отрицателно число. Диапазонът на числата, които могат да се представят, е от $-(2^{(k-1)} - 1)$ до $(2^{(k-1)} - 1)$ за k бита. Един от недостатъците на този метод е, че не може да се представи уникално нула, тъй като има два начина за представяне на 0 (положителен и отрицателен нула).
- **Обратен код (1's Complement Representation):** В този метод отрицателните числа се получават чрез инвертиране на всички битове на положителното число. Диапазонът на числата е същият като при **правия код**, но този метод също има проблем с представянето на нула, тъй като има два различни начина за представяне на числото 0 (положителен и отрицателен нула).
- **Допълнителен код (2's Complement Representation):** Това е най-често използваното представяне в компютърните системи, тъй като е по-ефективно и дава по-лесни и надеждни операции с числа. В този метод отрицателните числа се получават чрез инвертиране на битовете и добавяне на 1 към резултата. Диапазонът на числата е от $-(2^{(k-1)})$ до $(2^{(k-1)} - 1)$ за k бита. Този метод елиминира проблема с представянето на нула и е по-ефективен за изпълнение на аритметични операции. {3., 4.}

2.5. Пример за пресмятане на число

Представяме числото -43.625 в 32-битов формат, който резервира 1 бит за знака, 15 бита за целочислената част и 16 бита за дробната част. Тогава числото -43.625 може да се представи по следния начин: {3., 4.}

- 0 за положителен знак или 1 за отрицателен знак.
- 000000000101011₂ – 15-битово бинарно представяне за числото 43.
- 1010000000000000₂ – 16-битово бинарно представяне за дробната част 0.625.

1	000000000101011	1010000000000000
Sign bit	Integer part	Fractional part

Забележка: Точното представяне на дробната част се получава като преобразуваме десетичните стойности в бинарен формат. В случая 0.625 в десетична система се преобразува в 1010 в двоичен формат, като всеки бит съответства на стъпка на прецизност (в този случай 1/2, 1/4, 1/8 и т.н.).

Предимството на фиксираната запетая е бързината на изчисленията (поради предварително зададената прецизност), но нейният недостатък е сравнително ограниченото представяне на стойности. За числата, които изискват повече от 32 бита, ще се наложи да се съхраняват неточно. Въпреки това, фиксираната запетая е много полезна в приложения, изискващи бързи изчисления с ограничени ресурси, като в цифрови процесори или при обработка на сигнали.

2.6. Минимални и максимални стойности

За даден формат с 32 бита (1 бит за знак, 15 бита за целочислената част и 16 бита за дробната част), минималната стойност, която може да бъде представена, е $2^{-16} \approx 0.000015$, а максималната стойност е $2^{15} + (1 - 2^{-16}) \approx 32768$. Разликата между тези стойности е 2^{-16}

С помощта на фиксираната запетая можем да преместим запетаята наляво или надясно, като използваме само целочисленото поле. Например, при увеличаване на целочислената част се намалява прецизността на дробната част и обратно. {3.}

2.7. *Предимства на фиксираната запетая*

Предимствата на фиксираната запетая включват:

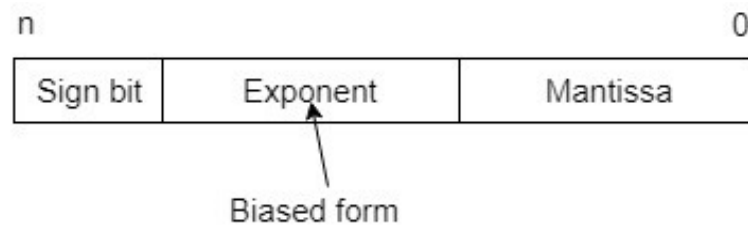
- **Бързина на изчисленията:** Поради фиксираната позиция на запетаята, изчисленията могат да се извършват много по-бързо, особено в хардуерни имплементации.
- **Лесно предсказуемо поведение:** Фиксираната запетая има предсказуемо поведение, което я прави подходяща за реално времеви приложения и системи с ограничени ресурси.
- **Малка изчислителна сложност:** При фиксираната запетая не е необходимо да се извършват сложни операции за управление на запетаята, което я прави по-проста за обработка от процесори с ограничени ресурси. Това е ключово предимство в системи с малка изчислителна мощност. {6.}

2.8. *Недостатъци на фиксираната запетая*

- **Ограничен диапазон:** Числата с фиксирана запетая не могат да обхванат много широк диапазон от стойности. Това може да бъде проблем при представянето на много големи или много малки числа.
- **Грешки при окръгляване:** Ако числата се представят с ограничен брой битове, може да възникнат грешки от окръгляване при операции с много малки или много големи числа. {6.}

3. Плаваща запетая

3.1. Представяне на числа с плаваща запетая



Представянето с плаваща запетая е система, при която не се резервира фиксиран брой битове за целочислената част или дробната част на числото. Вместо това, се резервират определен брой бита за числото (**мантиса**) и определен брой бита, които указват къде в това число се намира десетичната (или бинарната) запетая (**експонента**).

Числото, представено чрез плаваща запетая, има две основни части: първата част представлява знаковото фиксирано число, наречено мантиса. Втората част указва позицията на десетичната или бинарната запетая и е наречена експонента. Фиксираната мантиса може да бъде дробно или цяло число. Плаващата запетая винаги се интерпретира като число в следния вид: $M \times r^e$.

Всъщност, числото се изразява чрез формулата: $(-1)^{s(1+m)} \times 2^{(e-\text{Bias})}$, където **s** е битът за знак, **m** е мантисата, **e** е стойността на експонентата, а **Bias** е числото на изместването.

Трябва да се отбележи, че знаковите цели числа и експонентите могат да бъдат представени чрез представяне на знака, допълнение до 1 или допълнение до 2. {2., 3.}

3.2. Гъвкавост на плаващата запетая

Плаващата запетая е по-гъвкава, тъй като всяко ненулево число може да бъде представено в нормализиран вид като $\pm(1.b_1b_2b_3\ldots)_2 \times 2^n$. Това е нормализираната форма на числото x .

Пример:

Да разгледаме 32-битов формат: 1 бит за знака, 8 бита за експонентата и 23 бита за дробната част. Първоначалният бит 1 не се съхранява (тъй като винаги е 1 за нормализирани числа) и се нарича "скрит бит".

Пример за нормализиране на числото 53.5 $\rightarrow 53.5 = (-110101.1)_2 = (-1.101011) \times 2^5$.

Това ще бъде представено в следния вид:

- 00000101 – 8-битово бинарно представяне на експонента +5.

8-битовото поле за експонента се използва за съхранение на цели експоненти, като диапазонът е от -126 до 127.

1	00000101	101011000000000000000000
Sign bit	Exponent part	Mantissa part

3.3. Минимални и максимални стойности в плаващата запетая

Най-малкото нормализирано положително число, което може да бъде представено в 32 бита, е:

$$(1.000000000000000000000000)_2 \times 2^{-126} = 2^{-126} \approx 1.18 \times 10^{-38}.$$

Най-голямото нормализирано положително число е:

$$(1.111111111111111111111111)_2 \times 2^{127} = (2^{24} - 1) \times 2^{104} \approx 3.40 \times 10^{38}.$$

Тези стойности са представени по следния начин:

- Най-малката стойност: 000000000000000000000000 за мантиса, експонент: 00000000.
- Най-голямата стойност: 111111111111111111111111 за мантиса, експонент: 11111111.

{3.}

Smallest	0	10000010	000000000000000000000000
	Sign bit	Exponent part	Mantissa part
Largest	0	01111111	111111111111111111111111
	Sign bit	Exponent part	Mantissa part

3.4. Точност на плаващата запетая

Точността на плаваща запетая е броят на позициите, запазени за бинарни цифри, плюс една (за скрития бит). В разгледаните примери прецизността е $23 + 1 = 24$.

Разликата между 1 и следващото нормализирано число се нарича машинен епсилон. В примера по-горе, разликата е $(1 + 2^{-23}) - 1 = 2^{-23}$.

Това е същото като най-малкото положително число с плаваща запетая, тъй като разстоянието между числата не е равно, за разлика от ситуацията с фиксираната запетая. {3.,5.}

3.5. Ограничения на представянето с плаваща запетая

Заради ограниченото количество битове, с които се разполага за представяне на числата (например 32 или 64 бита за стандартни формати), не всички реални числа могат да бъдат точно представени с плаваща запетая. Например, числото $1/3$ има безкрайно двоично представяне (0.010101...), което не може да бъде представено точно. Това е характерно за всички числа, чието двоично представяне е безкрайно, като например $1/10$. {3., 4.}

3.6. Стандарт на IEEE за плаваща запетая

IEEE (Институт на електрическите и електронни инженери) е стандартизирал представянето на плаваща запетая по следния начин:

Числото се изразява с формулата: $(-1)^{s(1+m)} \times 2^{(e-Bias)}$, където s е битът за знак, m е мантисата, e е стойността на експонентата, а $Bias$ е изместването.

Според стандарта IEEE 754, плаващите числа се представят по следните начини:

- **Полу-прецизност (16 бита):** 1 бит за знак, 5 бита за експонента и 10 бита за мантиса.
- **Единична прецизност (32 бита):** 1 бит за знак, 8 бита за експонента и 23 бита за мантиса.
- **Двойна прецизност (64 бита):** 1 бит за знак, 11 бита за експонента и 52 бита за мантиса.

- **Четворна прецизност (128 бита):** 1 бит за знак, 15 бита за експонента и 112 бита за мантиса. {1.}

3.7. Специални стойности на IEEE 754

Според стандарта IEEE 754, има някои специални стойности, които зависят от различните стойности на експонентата и мантисата:

- **Всички експонентни битове 0 и всички мантисни битове 0** представляват 0. Ако битът за знак е 0, то числото е +0, ако е 1, то е -0.
- **Всички експонентни битове 1 и всички мантисни битове 0** представляват безкрайност. Ако битът за знак е 0, то числото е $+\infty$, ако е 1, то е $-\infty$.
- **Всички експонентни битове 0 и мантисни битове ненулеви** представляват денормализирано число.
- **Всички експонентни битове 1 и мантисни битове ненулеви** представляват грешка (NaN - Not a Number). {1.}

Примери за такива стойности:

- **+0:** 0000000000000000 (поле за експонента и мантиса са нули).
- **$+\infty$:** 0111111111110000 (експонент и мантиса с всички битове 1).
- **NaN (Not a Number):** 0111111111110001 (експонент с всички битове 1, мантиса ненулева).

3.8. Предимства на плаващата запетая

- **Гъвкавост:** Позволява представяне на много широк диапазон от стойности, включително много големи и много малки числа.

- **Динамична точност:** Предоставя висока точност за числа с голямо разнообразие от стойности, но точността може да варира в зависимост от размера на числата (например, малки числа могат да имат по-ниска точност поради ограничената мантисна част).
- **Поддръжка на специални стойности:** Включва представянето на специални стойности като 0, безкрайност ($\pm\infty$) и NaN (не е число).
- **Широко приет стандарт (IEEE 754):** Осигурява съвместимост между различни платформи и приложения. {6.}

3.9. Недостатъци на плаващата запетая

- **Неравномерно разстояние между числата:** При малки числа може да има по-малка точност, докато за много големи числа точността може да бъде загубена поради ограниченото количество битове за мантисата.
- **По-бавни изчисления:** Операциите с плаваща запетая изискват повече изчислителни ресурси и по-комплексни алгоритми за обработка, което ги прави по-бавни в сравнение с операциите с цели числа или с фиксирана запетая. {6.}

4. Сравнение на фиксирана и плаваща запетая

{6.}

Критерий	Фиксирана запетая	Плаваща запетая
Представяне на числата	Използва фиксиран брой битове за целочислената и дробна част	Числото се представя чрез мантиса и експонента с плаваща запетая
Гъвкавост	Ограничена гъвкавост, подходяща за определени диапазони от стойности	Много по-гъвкава, подходяща за широк диапазон от стойности (големи и малки числа)

Прецизност/Точност	Прецизността е ограничена до броя на битовете	Точността зависи от броя на битовете в мантисата и е по-голяма за плаващата запетая
Окръгляване	Поради ограничения брой битове възникват грешки при окръгляване	Възможни са грешки от окръгляване поради ограниченото количество на битове за мантиса
Представяне на специални стойност	Няма специални стойности като безкрайност или NaN	Поддържа специални стойности като 0, $+\infty$, $-\infty$ и NaN
Изчислителна сложност	По-бързи при операции с фиксирана запетая	Изчисленията с плаваща запетая са по-бавни поради нужда от обработката на мантисата и експонентата.
Използване	Подходяща за контролни системи и вградени приложения, където е нужно стриктно ограничение на прецизността	Използва се в научни изчисления, графика и области с широк диапазон от числа.

5. Заключение

Машинната аритметика и методите за представяне на числа играят решаваща роля за ефективността, прецизността и точността на изчисленията в компютърните системи. Изборът между представяне с фиксирана и плаваща запетая зависи от конкретните изисквания на приложението и изчисленията. Фиксираната запетая предлага по-бързи изчисления и по-малка изчислителна сложност, като е предпочитана в среди с ограничени ресурси, но с определени ограничения в диапазона на представените стойности и прецизността. От друга страна, плаващата запетая предоставя по-голяма гъвкавост и по-висока прецизност за широк диапазон от стойности, но с цената на по-бавни изчисления. Разбирането на тези два метода и тяхното прилагане е от съществено значение за оптимизацията на изчисленията в съвременните технологични системи, особено в контекста на научни изчисления, графика и системи с високоскоростни изисквания за обработка на данни.

Литература

1. IEEE Standard for Floating-Point Arithmetic (IEEE 754-2008).
2. Goldberg, D. (1991). *What Every Computer Scientist Should Know About Floating-Point Arithmetic*. ACM Computing Surveys.
3. TutorialsPoint – Fixed Point and Floating Point Number Representation
4. INFLIBNET Centre – Fixed and floating point arithmetic
5. WeDoLow – Floating-point or fixed-point: a choice between precision and efficiency
6. DEV Community – Floating-Point VS Fixed-point Arithmetic in Computing