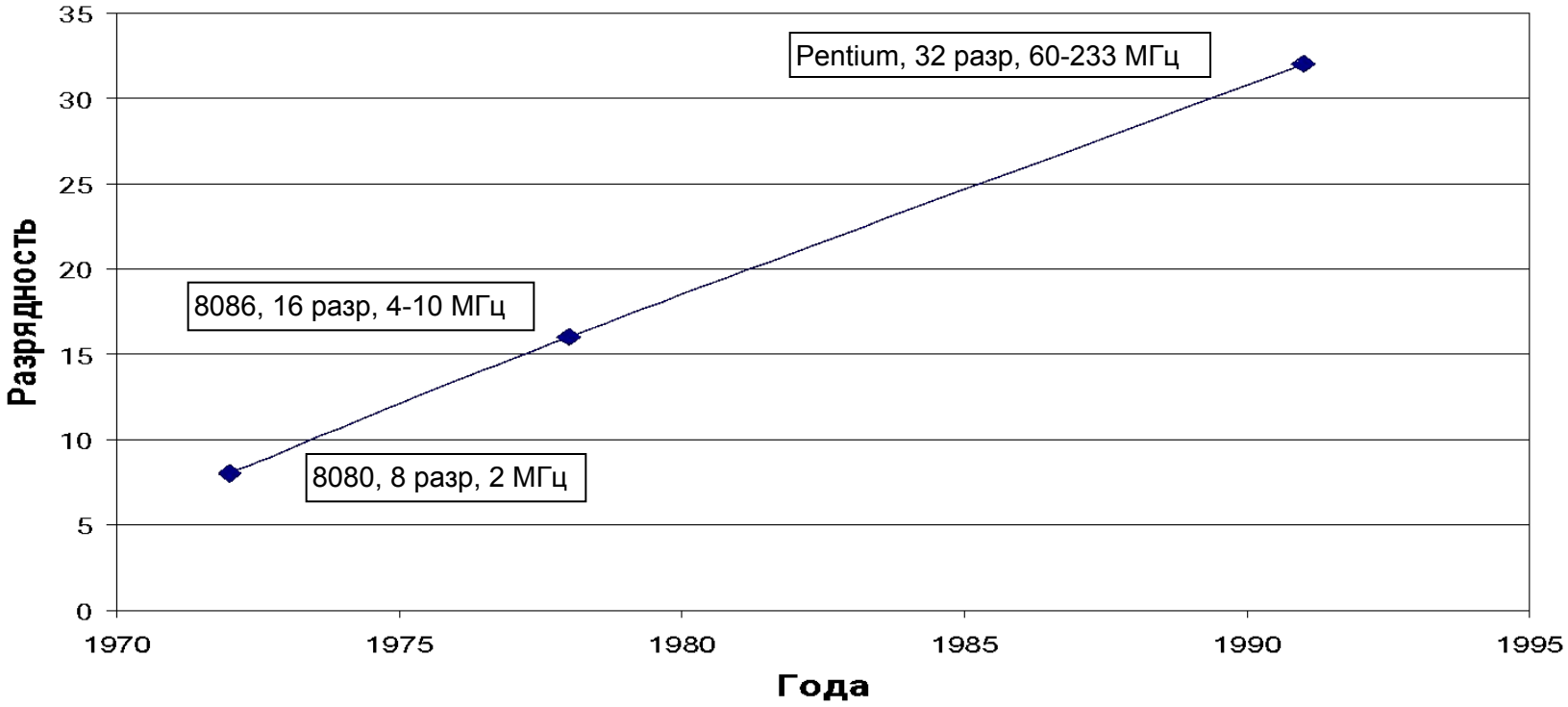
Лекция №4
по курсу
«Машинная арифметика в рациональных чисел»

Москва, 2021

Рост разрядности и тактовой частоты процессоров по годам



Гипотеза: Технологические трудности создания процессоров высокой разрядности

Вычитание близких друг другу по величине чисел

Рассмотрим два числа

$$x = 3.141592653589793$$

и

$$y = 3.141592653585682$$

Первое число, x является 16-значным приближением к числу пи, в то время как второе число соответствует 12-значному приближению к пи.

Их разница равна

$$z = x - y = 0.000000000004111 = 4.111 \cdot 10^{-12}$$

Аппроксимация производной

Отличной иллюстрацией потери точности является пример вычисления приближения к производной. Пусть f - непрерывно дифференцируемая функция одной действительной переменной, то есть такая, для которой производная f ' существует и является непрерывной.

Предположим, что у нас нет формулы для f, а есть только программа, которая оценивает f (x) для любого данного значения x. Тогда

$$\frac{f(x+h)-f(x)}{h} \qquad (11.4)$$

при $h->0$

Вычитание близких друг другу по величине чисел

```
int n = 1;
 double x = 1.0, h = 1.0;
 double deriv = Math.Cos(x), diffquo, error;
Console.WriteLine(deriv);

 while(n <= 20) {
    h = h / 10;
    diffquo = (Math.Sin(x + h) - Math.Sin(x)) / h; /* производная */
    error = Math.Abs(deriv - diffquo);
    //Console.WriteLine("h={0},diff={1}, error={2}",h, diffquo, error);
    Console.WriteLine(error);
    n++;
 }
```

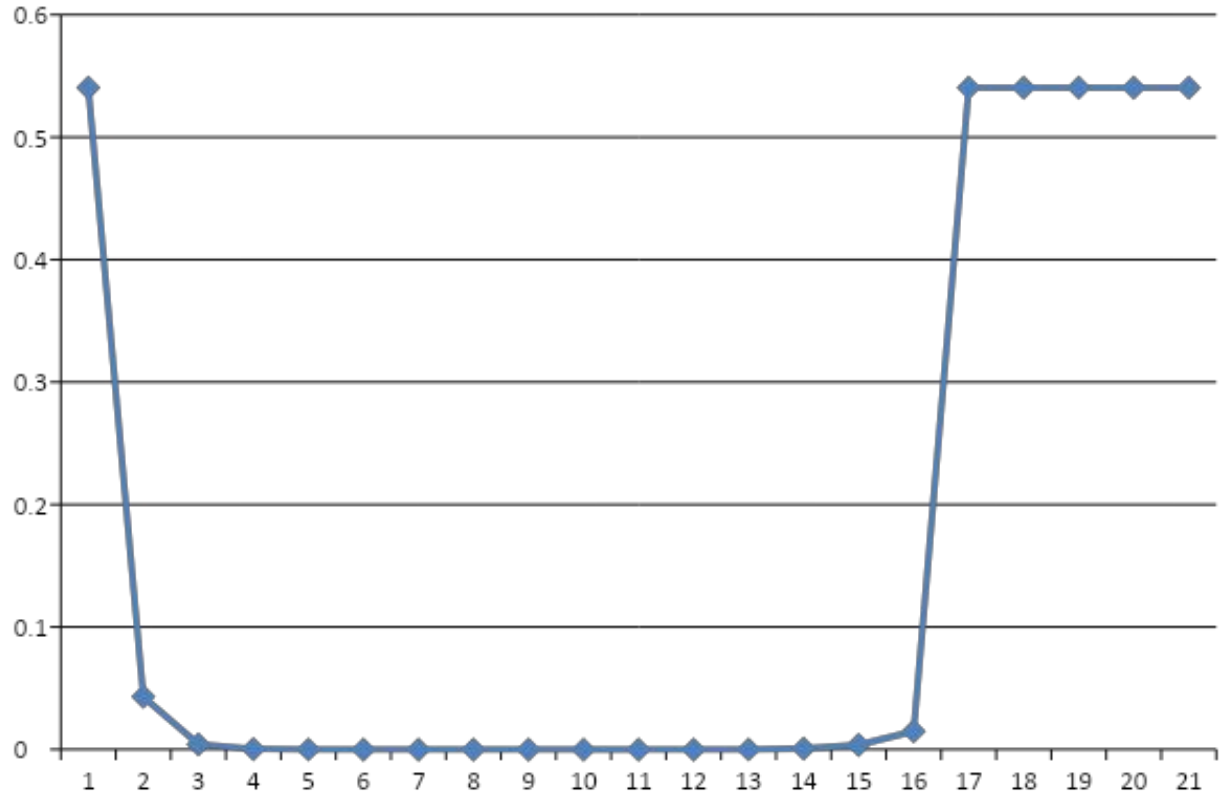
Вычитание близких друг другу по величине чисел

```
 deriv = 5.403023e-01
   h       diffquo   abs(deriv - diffquo)
1.0e-01  4.973638e-01  4.293855e-02
1.0e-02  5.360860e-01  4.216325e-03
1.0e-03  5.398815e-01  4.208255e-04
1.0e-04  5.402602e-01  4.207445e-05
1.0e-05  5.402981e-01  4.207362e-06
1.0e-06  5.403019e-01  4.207468e-07
1.0e-07  5.403023e-01  4.182769e-08
1.0e-08  5.403023e-01  2.969885e-09
1.0e-09  5.403024e-01  5.254127e-08
1.0e-10  5.403022e-01  5.848104e-08
1.0e-11  5.403011e-01  1.168704e-06
1.0e-12  5.403455e-01  4.324022e-05
1.0e-13  5.395684e-01  7.339159e-04
1.0e-14  5.440093e-01  3.706976e-03
1.0e-15  5.551115e-01  1.480921e-02
1.0e-16  0.000000e+00  5.403023e-01
1.0e-17  0.000000e+00  5.403023e-01
1.0e-18  0.000000e+00  5.403023e-01
1.0e-19  0.000000e+00  5.403023e-01
1.0e-20  0.000000e+00  5.403023e-01
```

x = 1.0, h = 1.0

Использование слишком большого h дает большую ошибку дискретизации, для малых h потерю точности. Для функции f (x) = sin (x) при x = 1,
лучший выбор h составляет примерно 10 ~ 8, приблизительно квадратный корень машинного эпсилона.

Вычитание близких друг другу по величине чисел



Аппроксимация производной

Для больших значений h ошибка уменьшается примерно в 10 раз каждый раз, когда h уменьшается в 10 раз до потери точности из-за ошибок округления

$$f(x+h) = f(x) + h \cdot f'(x) + \frac{h^2}{2} \cdot f''(z),$$

или

$$\frac{f(x+h) - f(x)}{h} - f'(x) = \frac{h}{2} \cdot f''(z),$$

Ошибка дискретизации O (h)

Аппроксимация производной

Центральная разность

Более точное приближение даёт формула:

$$\frac{f(x+h)-f(x-h)}{2\cdot h}=f'(x)$$

Это называется центральной разностью. Центральная разность дает лучшее приближение производной чем формула () так как:

$$f(x+h)=f(x)+h\cdot f'(x)+\frac{h^2}{2}\cdot f''(x)+\frac{h^3}{6}\cdot f'''(z_1),$$

для некоторого z_1 между x и x+h.

и

$$f(x-h)=f(x)-h\cdot f'(x)+\frac{h^2}{2}\cdot f''(x)-\frac{h^3}{6}\cdot f'''(z_2),$$

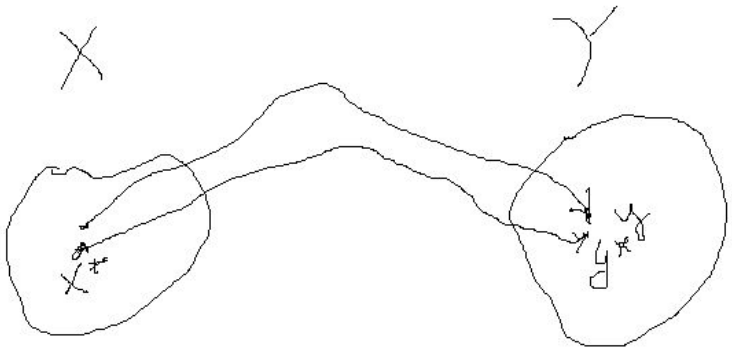
для некоторого z_2 между x и x-h.

Вычитая последние два выражения приходим к формуле:

$$\frac{f(x+h)-f(x-h)}{2\cdot h}-f'(x)=\frac{h^2}{12}(f'''(z_1)+f'''(z_2))$$

Это дает ошибку дискретизации для центральных разностей O (h^2) вместо O (h).

Обусловленность вычислительных задач

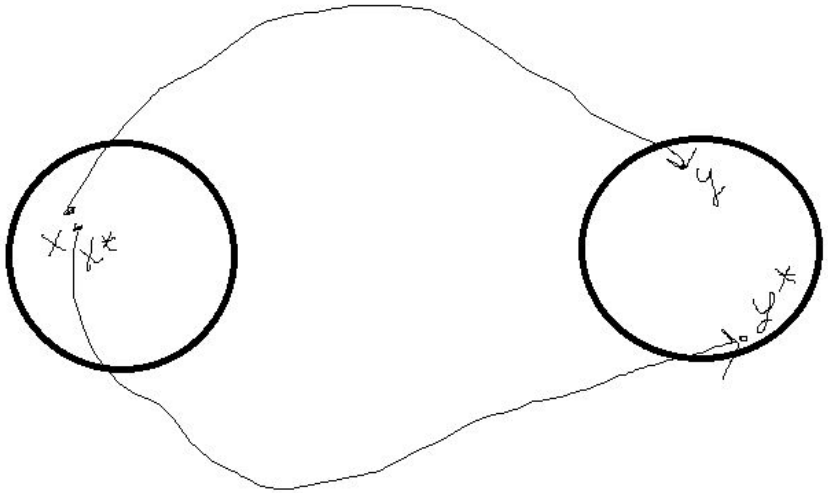


$\bar{x} \in X \qquad x^* = \bar{x} + \Delta\bar{x} \in X$

$\exists \bar{y} \in Y \qquad \exists\, y^*$;

y* будет мало отличаться от y

хорошо обусловленная задача



$\|x^* - x\| = \|\Delta x\|$ – мало

$\|y^* - y\|$ – велико

плохо обусловленная задача

Обусловленность вычислительных задач

$$y = f(x)$$

Пусть

f – дважды дифференцируемая функция и $x, f(x)$ находятся в диапазоне представления формата с плавающей точкой. Определим

$$\hat{x} = round(x)$$

Оценивая функцию f на компьютере, используя арифметику с плавающей запятой, лучше ~~всего~~ на что можно надеяться это рассчитать значение

$$\hat{y} = f(\hat{x})$$

На самом деле, даже это неоправданная надежда, потому что мы не сможем оценить точно f, но для простоты, давайте пока предположим, что это возможно. Известно, что относительная ошибка округления удовлетворяет неравенству:

$$-\log_{10}\left(\frac{|\hat{x}-x|}{|x|}\right) > -\log_{10}(\epsilon) \qquad \frac{|\hat{x}-x|}{|x|} < \varepsilon \qquad -\log_{10}\left(\frac{|\hat{y}-y|}{|y|}\right) \ ?$$

Обусловленность вычислительных задач

Имеем

$$\frac{\hat{y}-y}{y}=\frac{f(\hat{x})-f(x)}{\hat{x}-x}\cdot\frac{x}{f(x)}\cdot\frac{\hat{x}-x}{x}$$

Первый сомножитель

$\frac{f(\hat{x})-f(x)}{\hat{x}-x}$ аппроксимирует производную функции $f'(x)$

Следовательно,

$$\frac{|\hat{y}-y|}{|y|}\approx k_f(x)\times\frac{\hat{x}-x}{x},$$

где

$$k_f(x)=\frac{|x|\cdot|f'(x)|}{|f(x)|}$$

Величина kf (x) называется числом обусловленности f в точке x. Она измеряет приблизительно насколько увеличивается относительная ошибка округления в x при вычислении f(x).

Обусловленность вычислительных задач

$$-\log_{10}\left(\frac{|\hat{y}-y|}{|y|}\right) \approx -\log_{10}\left(\frac{|\hat{x}-x|}{|x|}\right) - \log_{10}\left(\kappa_f(x)\right)$$

Полезное эмпирическое правило.

Чтобы оценить количество цифр, с которым согласуется y^ = f (x^) с y = f (x)

требуется вычесть:

$\log_{10}(k_f(x))$

от приблизительного количества цифр, с которым x = round (x) совпадает с x, т. е.7 при использовании одинарной точности IEEE или 16 при использовании двойной точности IEEE. Предполагается, что f дважды непрерывно дифференцируема и что x и f (x) находятся в нормализованном диапазоне системы с плавающей точкой.

Поскольку оценка числа обусловленности f в x требует сначала оценки производная f '(x), число обусловленности не помогает нам решить нашу исходную проблема вычисления f(x). Оценка числа условия

сложнее, чем оценка функции, однако дает нам понимание трудностей, которые могут возникнуть, когда мы оцениваем f при определенных значениях x.

Пример обусловленности

Упражнение

Определить число обусловленности функции

$g(x) = x/10$, $h(x) = x - 10$

и обсудить для каких x, если таковые имеются, $k_g(x)$ или $k_h(x)$ велики.

Плохо обусловленная система

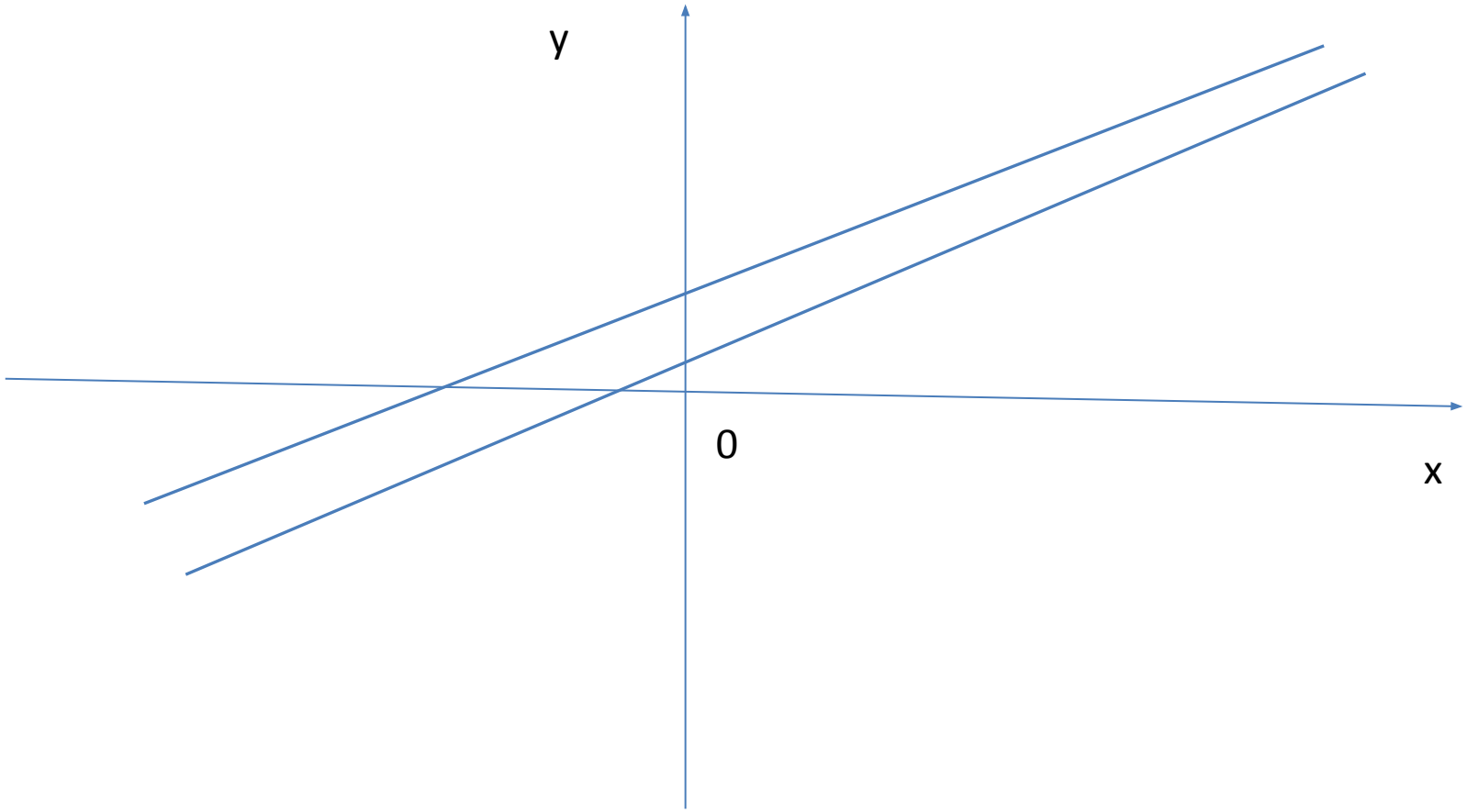
$$\begin{cases} 5x - 331y = 3.5 \\ 6x - 397y = 5.2 \end{cases}$$

x = 331.7; y = 5.000

$$\begin{cases} 5x - 331y = 3.5 \\ 6x - 397y = 5.1 \end{cases}$$

x = 298.6; y = 4.5

Плохо обусловленная система



Обусловленность

$$\lg \frac{\left|\hat{y} - y\right|}{|y|} \approx -\lg \frac{\left|\hat{x} - x\right|}{|x|} - \lg k_f(x),$$

Здесь левая часть - это примерно число цифр, с которыми y^ и y совпадает, а первый член в правой части равен примерно числу цифр, которые совпадают у x и x^ и равно 7 для формата с плавающей точкой одинарной точности.

Следовательно, приходим к следующему правилу.

Оценка количества ~~цифр~~ в которых

$$\hat{y} = f(\hat{x})$$

совпадает с

$$y = f(x)$$

равна разности количества ~~цифр~~ которых совпадают x и x^ (7 для одинарной точности и 16 для двойной) и

$$\lg k_f(x),$$

Упражнения

Упражнение 1

Определите число обусловленности для функции

$$f(x) = \frac{1}{x}$$

Существуют ли конечные ненулевые числа x, для которых f имеет наибольшее число обусловленности?

Упражнение 2

Составьте программу для определения числа обусловленности для формулы параллельного сопротивления с переменным сопротивлением R_1 и фиксированным $R_2 = 1$, то есть число обусловленности для функции:

$$f(x) = \frac{1}{\frac{1}{x} + 1}$$

Обусловленность

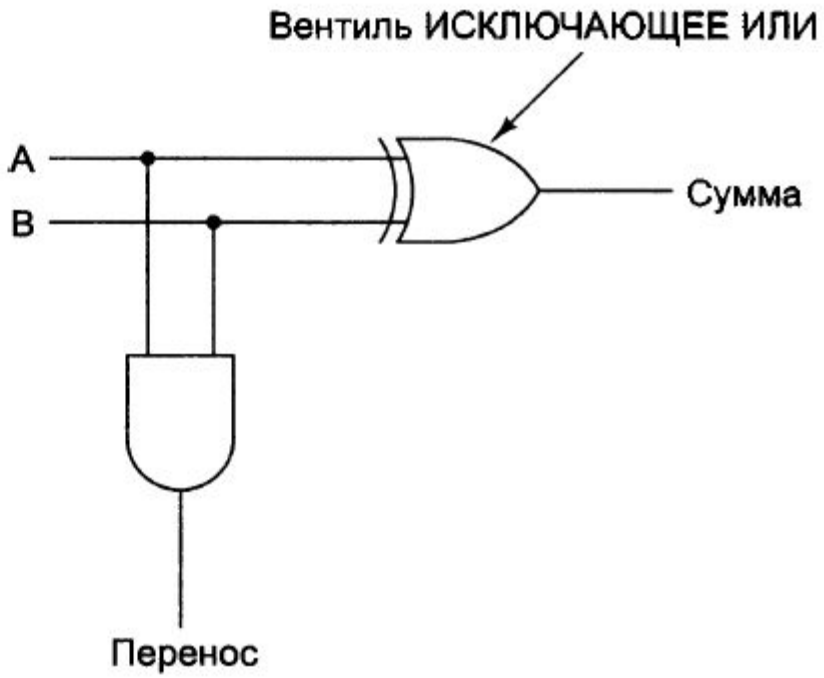
Задача считается хорошо обусловленной, если небольшие погрешности операндов приводят к небольшим изменением результатов и наоборот плохо обусловленной, если малые изменения исходных данных вызывает резкие изменения результатов. Оценка обусловленности задачи иногда в литературе называют анализом чувствительности.

Обратный анализ ошибок округления имеет отношение и применяется к алгоритму решения задачи, а анализ чувствительности к самой задаче.

Алгоритм решения вычислительной задачи обратно устойчив, если результат является точным решением слабо возмущенной задачи, т.е. обратно устойчивый алгоритм способен достаточно точно решать хорошо обусловленные задачи.

Полусумматор

A	B	Сумма	Перенос
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

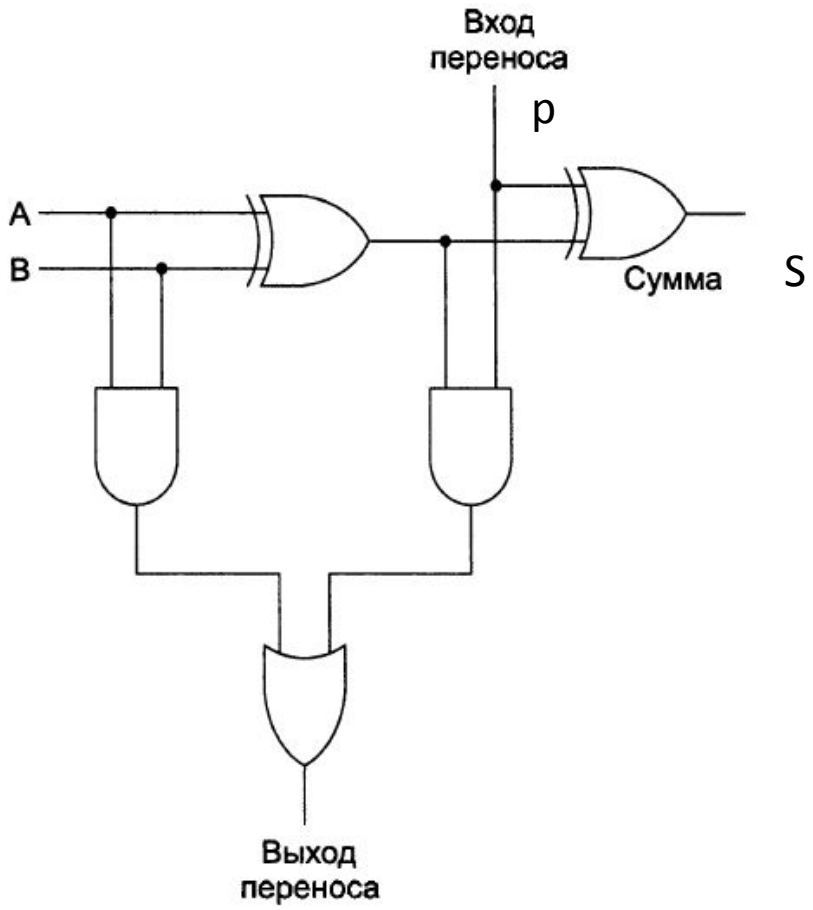


Сумматор со сквозным переносом

S = A+B+p

A	B	Вход переноса	Сумма	Выход переноса
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

а



б

+11111
 11111
_____

Сумматор с выбором переноса

Рассмотрим пример более быстрого сумматора. Разобьем 32-разрядный сумматор на 2 половины: нижнюю 16-разрядную и верхнюю 16-разрядную. Когда начинается сложение, верхний сумматор еще не может приступить к работе, поскольку не знает значение переноса, а узнать его он не сможет, пока не совершится 16 суммирований в нижнем сумматоре.

Однако можно сделать одно преобразование. Вместо одного верхнего сумматора можно получить два верхних сумматора, продублировав соответствующую часть аппаратуры. Тогда схема будет состоять из трех 16-разрядных сумматоров: одного нижнего и двух верхних, U_0 и U_1, работающих параллельно. В качестве переноса в сумматор U_0 поступает 0, в сумматор U_1 — 1. Оба верхних сумматора начинают работать одновременно с нижним сумматором, но только один из результатов суммирования в двух верхних сумматорах будет правильным. После сложения 16 нижних разрядов становится известно значение переноса в верхний сумматор, и тогда можно определить правильный ответ. При подобном подходе время сложения сокращается в два раза. Такой сумматор называется **сумматором с выбором переноса**. Можно еще раз разбить каждый 16-разрядный сумматор на два 8-разрядных и т. д.