Подходы к повышению производительности расширенного алгоритма Евклида для деления больших чисел двойной точности на большие числа одинарной точности

Национальный авиационный университет

Аспирант кафедры БИТ: Мария Ковтун

Руководитель: доцент кафедры БИТ, к.т.н. Сергей Гнатюк

Содержание

- Введение
- Актуальность
- Существующие алгоритмы
- Недостатки алгоритма прототипа
- Методы повышения производительности PAE
- Сравнение вычислительной сложности
- Сравнение производительности
- Выводы

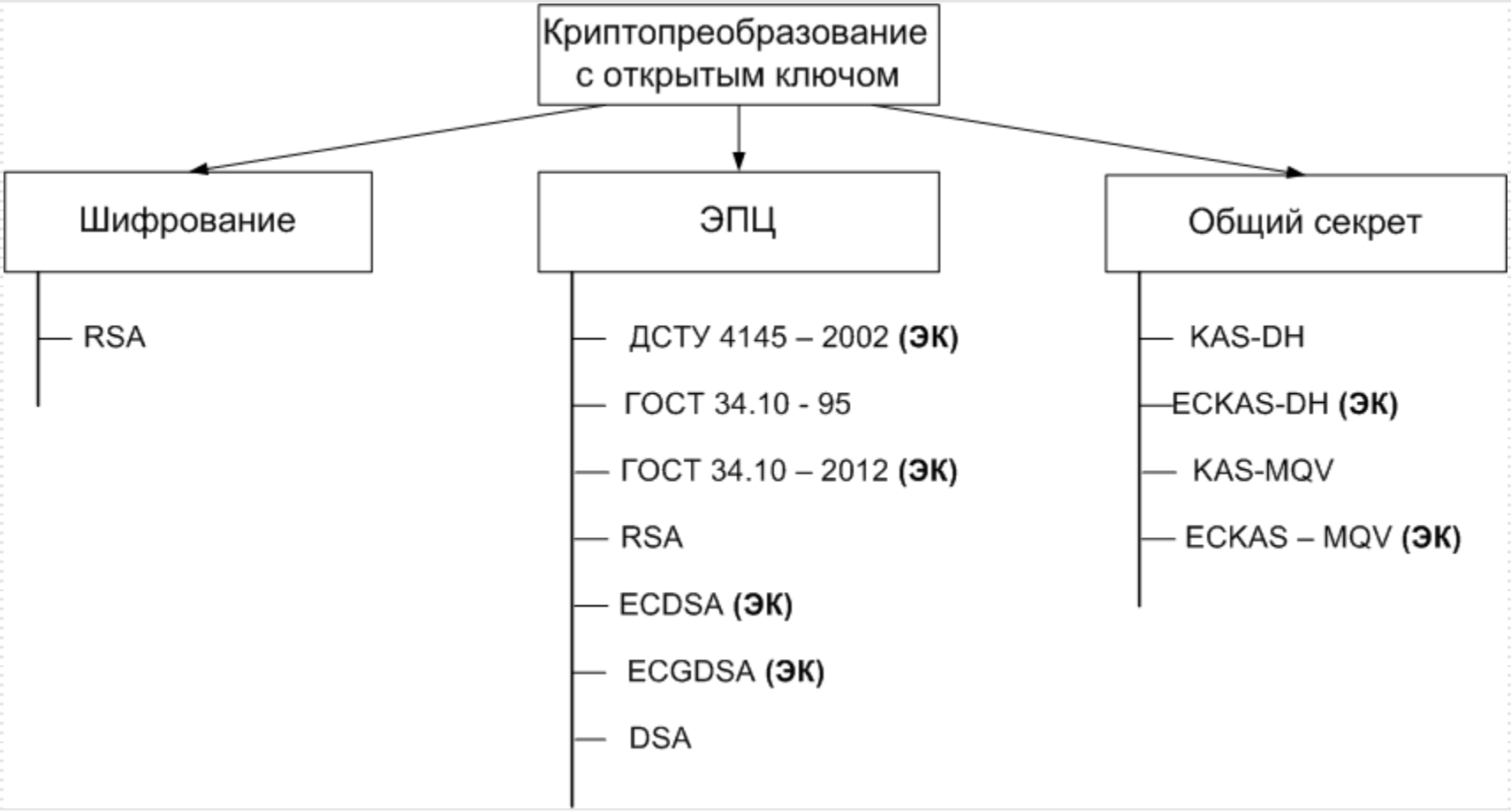
Введение

Цель: повышение производительности операции деления больших целых чисел с отличающейся в 2 раза двоичной длиной

Объект: операция деления больших целых чисел

Предмет: расширенный алгоритм Евклида

Актуальность



Актуальность

<table>
<tr><td>Криптопреобразования</td><td colspan="3">Зашифровывание/ расшифровывание</td><td colspan="3">Формирование и проверка цифровой подписи</td><td colspan="2">Обмен ключами</td></tr>
<tr><td rowspan="2">Арифметика в группе точек эллептической кривой</td><td colspan="5">Скалярное умножение точек эллиптической кривой</td><td colspan="2" rowspan="2">Генерация случайной точки</td></tr>
<tr><td colspan="3">Сложение точек</td><td colspan="2">Удвоение точки</td></tr>
<tr><td>Арифметика в поле GF(p)</td><td>Умноже-ние</td><td>Сложе-ние</td><td>Деление</td><td>Возведение в квадрат</td><td>Приведе-ние по модулю</td><td>Инверти-рование</td><td>Извлечения квадратного корня</td></tr>
<tr><td>Операции над массивами</td><td colspan="2">Сдвиг</td><td>Сравнение</td><td>Сложение</td><td>Вычитание</td><td>Умножение</td></tr>
<tr><td>Команды CPU</td><td colspan="7">mov, mul, shr, shl, add, sub</td></tr>
</table>

Известные алгоритмы

- Классический («деление в столбик»)
- Барретта и его модификации
- Монтгомери и его модификации
- Расширенный алгоритм Евклида
- Деление Джебелина
- Итераций Ньютона

Повышение производительности

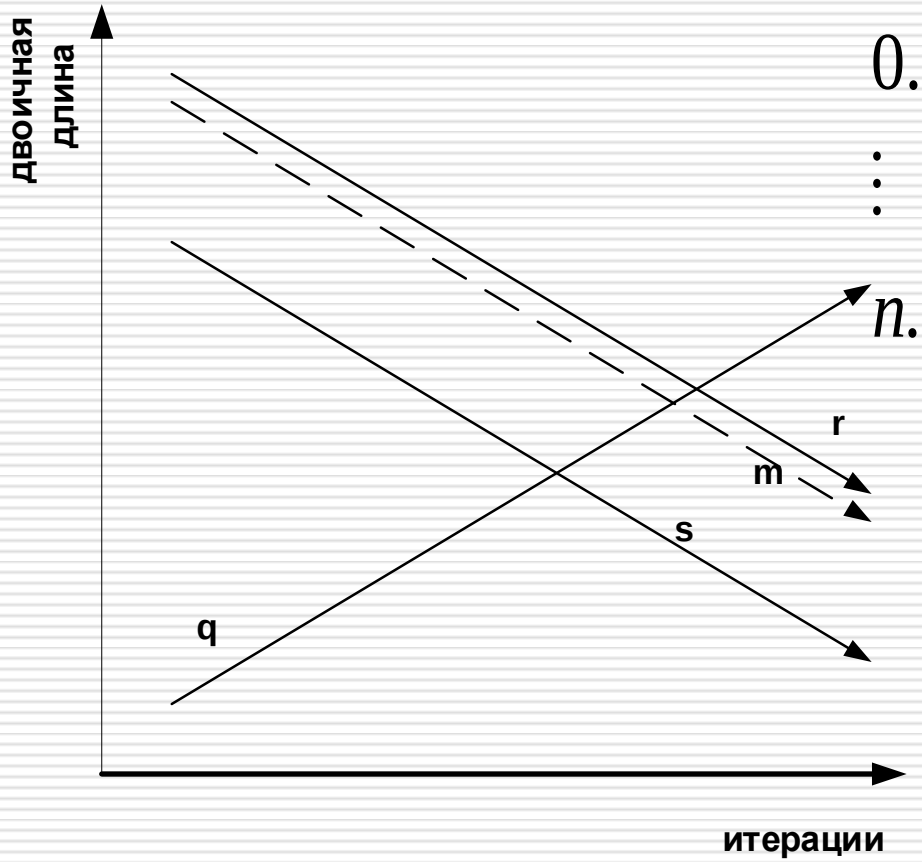
- Увеличение разрядности машинных слов
- Использование специализированных потоковых команд процессоров (MMX, SSE2, AVX, CLMUL, AVX2, ADOX*, ADCX*)
- Распараллеливание (многопоточность)
- Совершенствование алгоритмов
- Совершенствование структур данных

* - поддерживаются процессорами компании Intel, с января 2015 года.

Алгоритм прототип

РАСШИРЕННЫЙ АЛГОРИТМ ЕВКЛИДА ДЛЯ ДЕЛЕНИЯ БОЛЬШИХ ЦЕЛЫХ ЧИСЕЛ

РАЕ: Изменение параметров уравнения Безу



$$0.\ b \cdot q_0 + r_0 \cdot \alpha_0 = a \,|\, q_0 = 0, r_o = a$$

$$\vdots$$

$$n.\ b \cdot q + r \cdot \alpha = a \,|\, \alpha = 1$$

РАЕ: Алгоритм

Алгоритм 1. Расширенный алгоритм Евклида для деления больших целых чисел, когда двоичная длина делимого в два раза превышает длину делителя.

Вход: $a,b \in \mathbf{Z}$, $a,b \neq 0$, $n_a \leftarrow \left\lceil \frac{\log_2 a}{w} \right\rceil$, где n_a - количество машинных слов, которое занимает делимое; $n_b \leftarrow \left\lceil n_a / 2 \right\rceil$ - количество машинных слов, которое занимает делитель; w - ширина машинного слова, обычно $w=32$.

Выход: $q,r \in \mathbf{Z}$.

1. $r \leftarrow a$, $m \leftarrow b$, $s \leftarrow 1$, $q \leftarrow 0$.

2. While $r > m$ do

2.1 $m \leftarrow m << 1$, $s \leftarrow s << 1$.

3. While $r > b$ do

3.1. While $r < m$ do

3.1.1. $m \leftarrow m >> 1$, $s \leftarrow s >> 1$.

3.2. $r \leftarrow r - m$, $q \leftarrow q + s$.

4. Return (r,q).

РАЕ: Недостатки

- Вычислительно сложная проверка при сравнении больших чисел. При сравнении анализируются все машинные слова.
- После каждой итерации сравнения происходит сдвиг влево или в право лишь на 1 бит, когда двоичная длина делимого и делителя весомая.
- При сдвиге машинных слов сдвигаются все машинные слова, даже заведомо нулевые.
- При вычитании используются все машинные слова, даже нулевые, когда уменьшаемое меньше вычитаемого.

РАЕ: Сложность

$$I(A_1) = 5 \cdot k \left\lceil \frac{n}{w} \right\rceil L + C \cdot k \left(\frac{1}{3} \left\lceil \frac{n}{w} \right\rceil + \left\lceil \frac{n - n_a}{w} \right\rceil + \left\lceil \frac{n - n_b}{w} \right\rceil \right)$$

где

L - арифметические операции;

C - операции сравнения;

n_a - номер старшего бита делимого;

n - количество бит, зарезервированных в памяти для хранения максимального возможного числа,

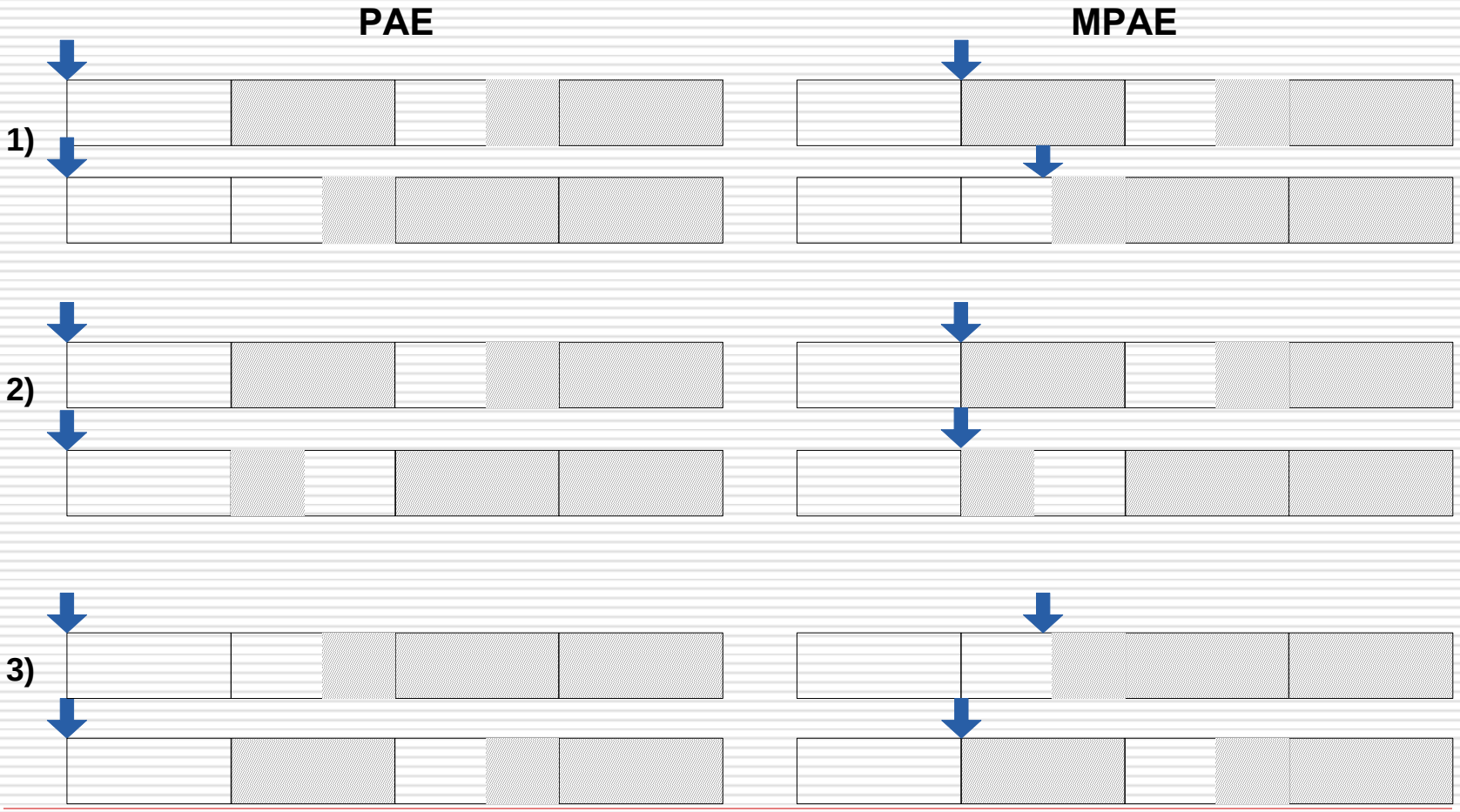
k - разница между номером старшего бита делимого и делителя;

w - ширина машинного слова.

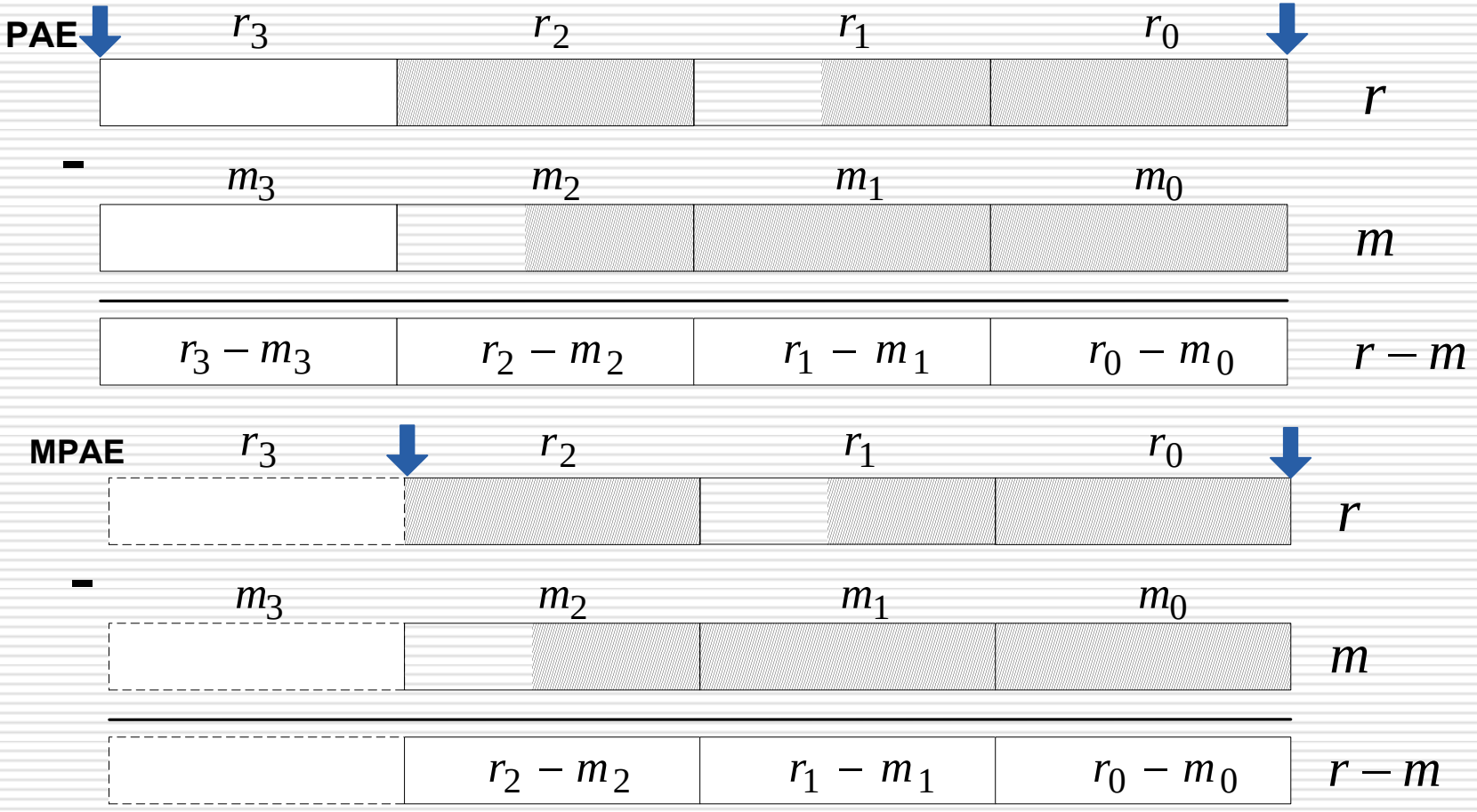
Предложенный алгоритм

МОДИФИЦИРОВАННЫЙ РАСШИРЕННЫЙ АЛГОРИТМ ЕВКЛИДА ДЛЯ ДЕЛЕНИЯ БОЛЬШИХ ЦЕЛЫХ ЧИСЕЛ

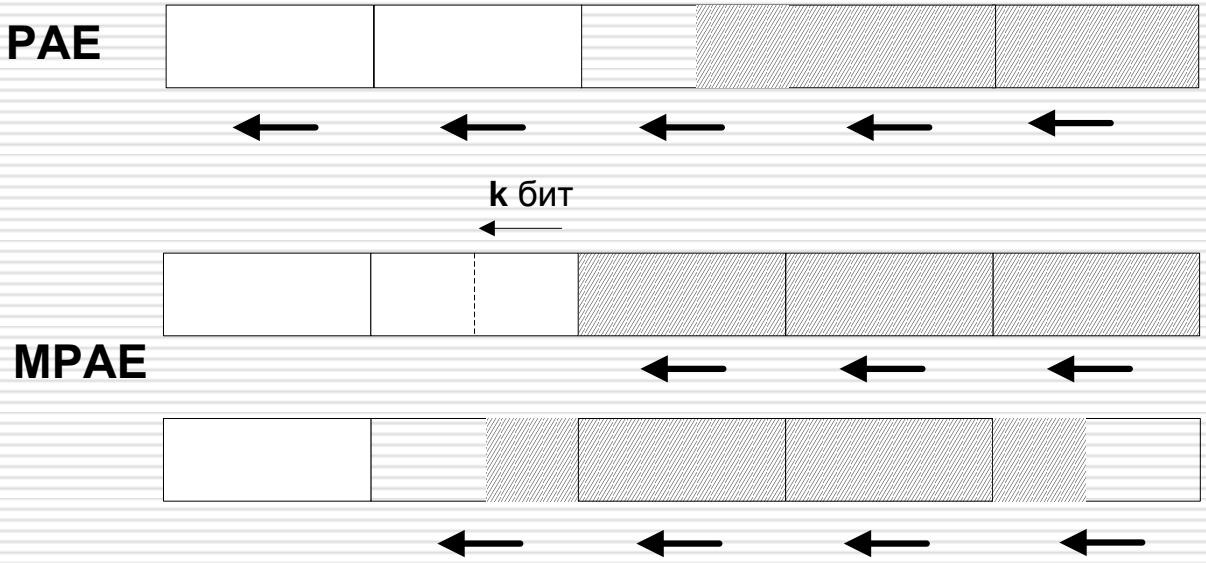
PAE и MPAE: Операция сравнения



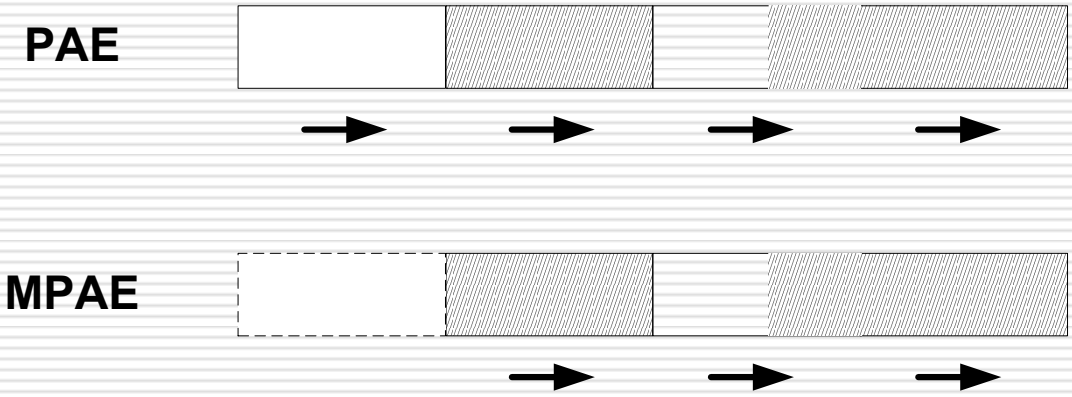
PAE и MPAE: Операция вычитания



PAE и MPAE: Операция сдвига влево



PAE и MPAE: Операция сдвига вправо



MPAE: Усовершенствования

- При сравнении больших целых чисел сравнивать номера старших битов, а в случае их равенства, проводить пословное сравнение значимых слов.
- Производить сдвиги за одну итерацию, зная разницу номеров старших битов, лишь значимых слов.
- Вычитать лишь значимые слова.

Алгоритм MPAE: Обозначения

$\tilde{m}$ - количество значимых слов числа;

M - номер старшего бита;

$sgf(\cdot)$ - функция вычисления количество значимых слов большого числа;

$msb(\cdot)$ - функция вычисления номера старшего бита большого числа;

$r \underset{\tilde{t}}{\geq} t$ - сравнение больших целых чисел по значимым словам;

$m \leftarrow m \underset{\tilde{m}}{<<} 1$ - сдвиг влево по значимым словам.

$r \leftarrow r \underset{\tilde{r}}{-} m$ - вычитание по значимым словам.

Алгоритм MPAE

Алгоритм 2. Модифицированный расширенный алгоритм Евклида для деления больших целых чисел.

Вход: $a,b \in \mathbf{Z}$, $a,b \neq 0$, $n_a \leftarrow \left\lceil \frac{\log_2 a}{w} \right\rceil$, где n_a - количество машинных слов, которое занимает делимое; $n_b \leftarrow \left\lceil n_a / 2 \right\rceil$ - количество машинных слов, которое занимает делитель; w - ширина машинного слова, обычно $w = 32$.

Выход: $q,r \in \mathbf{Z}$.

1. $r \leftarrow a$, $m \leftarrow b$, $s \leftarrow 1$, $q \leftarrow 0$, $t \leftarrow b$, $S \leftarrow 1$, $\tilde{s} \leftarrow 1$.
2. $M \leftarrow msb(m)$, $\tilde{m} \leftarrow sgf(m)$, $R \leftarrow msb(r)$, $\tilde{r} \leftarrow sgf(r)$ $T \leftarrow M$, $\tilde{t} \leftarrow \tilde{m}$.
3. $k \leftarrow R - M$.
4. if $k > 0$ then $m \leftarrow \underset{\tilde{m}}{m << k}$, $s \leftarrow \underset{\tilde{s}}{s << k}$, $M \leftarrow M + k$, $S \leftarrow S + k$, $\tilde{m} \leftarrow sgf(m)$.
5. While $(R > M) or \left(\underset{\tilde{m}}{r > m} \right)$ do

5.1. $\tilde{s} \leftarrow sgf(s)$.

5.2. $m \leftarrow \underset{\tilde{m}}{m << 1}$, $s \leftarrow \underset{\tilde{s}}{s << 1}$, $M \leftarrow M + 1$, $S \leftarrow S + 1$, $\tilde{m} \leftarrow sgf(m)$.

Алгоритм MPAE

6. While $(R>T) or \left(r \underset{\max(\tilde{r},\tilde{t})}{>} t \right)$ do

6.1. $k \leftarrow M - R$.

6.2. if $k>0$ then $\tilde{s} \leftarrow sgf(s)$, $\tilde{m} \leftarrow sgf(m)$.

6.3. $m \underset{\tilde{m}}{\leftarrow} m >> k$, $s \underset{\tilde{s}}{\leftarrow} s >> k$, $M \leftarrow M - k$, $S \leftarrow S - k$, $\tilde{m} \leftarrow sgf(m)$.

6.4. While $(R<M) or \left(r \underset{\max(\tilde{m},\tilde{r})}{<} m \right)$ do

6.4.1. $\tilde{s} \leftarrow sgf(s)$.

6.4.2. $m \underset{\tilde{m}}{\leftarrow} m >> 1$, $s \underset{\tilde{s}}{\leftarrow} s >> 1$.

6.4.3. $M \leftarrow M - 1$, $S \leftarrow S - 1$.

6.4.4. $\tilde{m} \leftarrow sgf(m)$.

6.5. $\tilde{r} \leftarrow sgf(r)$, $r \underset{\tilde{r}}{\leftarrow} r - m$, $R \leftarrow msb(r)$, $\tilde{r} \leftarrow sgf(r)$.

6.6. $q \leftarrow q + s$.

7. Return (q, r).

Алгоритм MPAE: Сложность

$$I(A_2) = L\left(63 + \left\lceil \frac{n_a}{w} \right\rceil \left(\frac{5}{3} + \frac{7}{12}k\right) + \left\lceil \frac{k}{w} \right\rceil \left(\frac{k}{3} + \frac{5}{3}\right) + \left\lceil \frac{n}{w} \right\rceil \frac{k}{2} + \frac{207}{12}k\right) + C\left(4 + \frac{5}{2}k\right)$$

где

L - арифметические операции;

C - операции сравнения;

n_a - номер старшего бита делимого;

n_b - номер старшего бита делителя;

n - количество бит, зарезервированных в памяти для хранения максимального возможного числа,

k - разница между номером старшего бита делимого и делителя;

w - ширина машинного слова.

PAE vs MPAE

СРАВНЕНИЕ

Рекомендованные длины ключей

Длина ключа	Симетрич-ные алгоритмы	Длина ключа RSA/DH	Длина ключа для Криптосистем на эллиптических кривых		Криптосистемы на гипперэллиптических кривых
			Простое $\mathbf{F}(p)$	Двоичное $\mathbf{F}(2^m)$	Двоичное $\mathbf{F}(2^m)$, g=2
80	2TDEA	1024	192	163	67
112	3TDEA	2048	224	233	97
128	AES-128	3072	256	283	117
192	AES-192	7680	384	409	171
256	AES-256	15360	521	571	237

MPAE vs PAE

СРАВНЕНИЕ ВЫЧИСЛИТЕЛЬНОЙ СЛОЖНОСТИ

Сравнение сложностей

ρ	1	0,9	0,8	0,7	0,6	0,5	0,4	0,3	0,2	0,1
$64\,бита \mid n_a = 2,\ 32\,бита \mid n_b = 1$										
k	29	32	35	39	42	45	48	51	55	58
$I_L(A_1)$	320	352	384	416	448	480	512	544	576	608
$I_L(A_2)$	701	788	860	932	1003	1075	1147	1219	1290	1362
$I_C(A_1)$	74	105	114	124	134	143	153	162	172	181
$I_C(A_2)$	77	85	93	101	109	117	125	133	141	149
$I_L(A_1)/I_L(A_2)$	0,46	0,45	0,45	0,45	0,45	0,45	0,45	0,45	0,45	0,45
$I_C(A_1)/I_C(A_2)$	0,97	**1,24**	**1,24**	**1,23**	**1,23**	**1,23**	**1,23**	**1,22**	**1,22**	**1,22**
$16384\,бита \mid n_a = 512,\ 8132\,бита \mid n_b = 256$										
k	8 162	8 981	9 800	10 620	11 439	12 258	13 077	13 896	14 716	15 535
$I_L(A_1)$	20 971 520	23 068 672	25 165 824	27 262 976	29 360 128	31 457 280	33 554 432	35 651 584	37 748 736	39 845 888
$I_L(A_2)$	6 423 899	7 180 839	7 963 161	8 761 471	9 585 982	10 431 792	11 292 361	12 180 360	13 082 299	14 012 487
$I_C(A_1)$	2 084 372	2 524 758	3 007 316	3 521 503	4 087 595	4 683 694	5 333 320	6 025 118	6 744 490	7 519 822
$I_C(A_2)$	20 409	22 457	24 505	26 553	28 601	30 649	32 697	34 745	36 793	38 841
$I_L(A_1)/I_L(A_2)$	**3,26**	**3,21**	**3,16**	**3,11**	**3,06**	**3,02**	**2,97**	**2,93**	**2,89**	**2,84**
$I_C(A_1)/I_C(A_2)$	**102,13**	**112,43**	**122,72**	**132,62**	**142,92**	**152,82**	**163,11**	**173,41**	**183,31**	**193,61**

MPAE vs PAE

СРАВНЕНИЕ ПРОИЗВОДИТЕЛЬНОСТИ

Программная реализация

- ОС: Microsoft Windows 7 pro x86-64
- Компиляторы: (Target – x86):
  - Microsoft Visual C++ 2010 (/O3, поддержка SSE2)

Эксперимент: условия

- Среднее для 100 тыс. операций.
- Вычислительные системы:
  - Intel Core i3 M350 (PC1)
  - Intel Xeon CPU E5 – 2640 (PC2)

Замеры производительности

	PC1	PC2	PC1	PC2	PC1	PC2	PC1	PC2	PC1	PC2	PC1	PC2
ρ	10%		20%		40%		50%		80%		90%	
t	**мс**											
$512\,бит \mid n_a = 16,\ 256\,бит \mid n_b = 8, w = 32$												
ω	224/ 12	224/ 12	224/ 23	224/ 23	224/ 42	224/ 42	224/ 54	224/ 54	224/ 97	224/ 97	224/ 111	224/ 111
div2*	0,049	0,039	0,047	0,033	0,041	0,030	0,040	0,030	0,030	0,025	0,030	0,022
div2	0,098	0,072	0,092	0,062	0,080	0,057	0,075	0,055	0,058	0,043	0,053	0,038
	2,0	**1,8**	2,0	**1,9**	2,0	**1,9**	1,9	**1,8**	1,9	**1,7**	1,8	**1,7**
$1024\,бита \mid n_a = 32, 512\,бит \mid n_b = 16, w = 32$												
ω	445/ 18	445/ 18	445/ 46	445/ 46	445/ 84	445/ 84	445/ 93	445/ 93	445/ 176	445/ 176	445/ 199	445/ 199
div2*	0,156	0,129	0,160	0,120	0,139	0,117	0,131	0,108	0,113	0,096	0,108	0,083
div2	0,380	0,277	0,360	0,256	0,309	0,235	0,289	0,210	0,229	0,172	0,209	0,152
	2,4	**2,1**	2,3	**2,1**	2,2	**2,0**	2,2	**1,9**	2,0	**1,8**	1,9	**1,8**

Замеры производительности

	PC1	PC2	PC1	PC2	PC1	PC2	PC1	PC2	PC1	PC2	PC1	PC2
ρ	10%		20%		40%		50%		80%		90%	
t	**мс**											
$4096\,бит \mid n_a = 128,\ 2048\,бит \mid n_b = 64, w = 32$												
ω	1696/81	1696/81	1696/170	1696/170	1696/334	1696/334	1696/413	1696/413	1696/705	1696/705	1696/774	1696/774
div2*	2,371	1,732	2,169	1,653	1,996	1,514	1,919	1,419	1,575	1,139	1,31	1,129
div2	6,224	4,305	5,803	4,072	5,133	3,572	4,805	3,276	3,791	2,590	3,37	2,32
	2,6	**2,5**	2,7	**2,5**	2,6	**2,4**	2,5	**2,3**	2,4	**2,3**	2,6	**2,1**
$16384\,бит \mid n_a = 512,\ 8192\,бита \mid n_b = 256, w = 32$												
ω	6969/314	6969/314	6969/691	6969/691	6969/1405	6969/1405	6969/1742	6969/1742	6969/2787	6969/2787	6969/3137	6969/3137
div2*	35,007	25,054	32,65	24,617	30,467	21,715	28,876	21,076	23,743	17,378	21,559	15,896
div2	96,299	63,944	90,683	60,668	79,685	53,165	74,474	49,452	58,359	39,484	53,025	36,099
	2,8	**2,6**	2,8	**2,5**	2,6	**2,4**	2,6	**2,3**	2,5	**2,3**	2,5	**2,3**

MPAE vs PAE

ВЫВОДЫ

Выводы

- Проведено исследование известных алгоритмов деления и выбран РАЕ в качестве прототипа.
- Проведен анализ РАЕ и выявлены его основные недостатки (слабые стороны).

Выводы

- Предложены основные подходы к улучшению производительности PAE:
  - Приближенное сравнение целых чисел.
  - Знания закона изменения параметров уравнения Безу, для вычисления числа значимых машинных слов.
  - Операции только над значимыми словами (вычитание, сдвиг и сравнение).

Выводы

- Сравнение аналитической средней оценка вычислительной сложности PAE и MPAE, показала выигрыш в **1,24-193,91** раз в количестве операций сравнения и выигрыш в **1,34-3,26** (число от 256 бит) раз в количестве арифметических операций, с ростом двоичной длины большого целого числа.

Выводы

- Экспериментальные оценки производительности программной реализации PAE и MPAE показали линейную зависимость от разницы двоичных длины делимого и делителя.

Выводы

- Предложенные подходы к оптимизации PAE для деления больших целых чисел, позволили повысить производительность программной реализации MPAE в **1,5-3** раза с ростом количества машинных слов.

Выводы

- РАЕ для деления больших целых чисел имеет ограниченную область применения при делителе намного меньшем делимого - время выполнения увеличивается в несколько раз. Становится сильно заметно даже на числах с небольшой длиной. В таких случаях лучше применять специальные версии алгоритмов.

Выводы

- Предложенный MPAE для деления больших целых чисел, не ориентирован на многопоточное выполнение, что не позволило полностью реализовать потенциал современных многоядерных процессоров.

Дальнейшие исследования

- Распараллеливание: многопоточное выполнение, для реализации потенциала современных многоядерных процессоров.
- При сложении и вычитании использовать арифметику «с отложенным переносом».

Вопросы?

Спасибо за внимание!

Национальный авиационный университет

Кафедра безопасности информационных технологий

Мария Ковтун

email: mg.kovtun@gmail.com