Предложения по реализации умножения в поле Галуа над неприводимым многочленом на примере преобразования *L* в алгоритме ГОСТ Р 34.12-2015

магистрант[1], инженер [2] Салимов Г. Ю.,
ПГУ[1], АО “ПНИЭИ”[2], г. Пенза.
salimovgorik@mail.ru

План исследования

1. Обзор существующих методов;
2. Математическое описание умножение в поле Галуа над неприводимым многочленом;
3. Имплементация на языке программирования высокого уровня;
4. Экспериментальное исследование на различных вычислительных системах.

Проанализированные реализации алгоритма шифрования

Автор	Источник	Тип реализации *L*
1. ТК 26	tc26.ru/standard/gost/PR_GOSTR_bch_v6.zip https://tc26.ru/standard/draft/PR_GOSTR-bch_v3.zip	Таблица
2. Маркку Сааринен (Markku-Saarien)	https://github.com/mjosaarinen/kuznechik	Алгоритм
3. Роман Олейников	https://github.com/Roman-Oliynykov/ciphers-speed	Таблица
4. Эрих Фийол (Eric Filiol)	https://drive.google.com/file/d/0B6BlkqAoxXq1bDJURGRhamtPb00/view?pli=1	Таблица
5. Thomas Aprelev @aprlv, C99 SSE2	https://github.com/aprelev/lg15	Таблица
6. Maxim Gorky @sebastian_mg	https://habr.com/ru/users/sebastian_mg/	Таблица
7. Максим Тишков	https://github.com/MaXaMaR/kuznezhik	Алгоритм
8. @stargrave2, Python	http://git.cypherpunks.ru/cgit.cgi/gogost.git/	Таблица
9. Aleksey Never, Python	https://github.com/NeverWalkAloner/Cryptography-standards	Алгоритм
10. Alexander Venedioukhin, GO	https://dxdt.ru/golang/gost/cipher/kuznec/kuznec.go	Алгоритм
11. Nikolai Kim @yaruson	https://github.com/yaruson/GostPlugin	Алгоритм
12. @ru_cryptf, подборка	https://habr.com/ru/post/273055/	Алгоритм/Таблица
13. Анонимная реализация	https://sourceforge.net/projects/cppcrypto/	Таблица
14.Эрей Бьянски (Erie Banksy)	https://github.com/ErieBanksy/GOST-R-34.12-2015	Алгоритм

Влияние реализации преобразования L на использование ресурсов вычислительной системы

Критерий	Алгоритмическая реализация (прямое вычисление произведения)	Табличная реализация (использование таблиц предвычислений)
Скорость	Низкая	**Высокая**
Расход памяти	**Низкий**	Высокий
Преимущества	•Низкое потребление памяти; •Возможность реализации на малоресурсных микроконтроллерах; •Независима от введения нового полинома, коэффициентов умножения.	Высокая скорость шифрования/дешифрования
Недостатки	Не высокая скорость шифрования/дешифрования	•Необходимо хранить таблицу предвычислений; •Отсутствие возможности реализации на малоресурсных микроконтроллерах; •Необходимо перепроектировать работу с таблицей при изменении полинома либо коэффициентов умножения.

Преобразование L

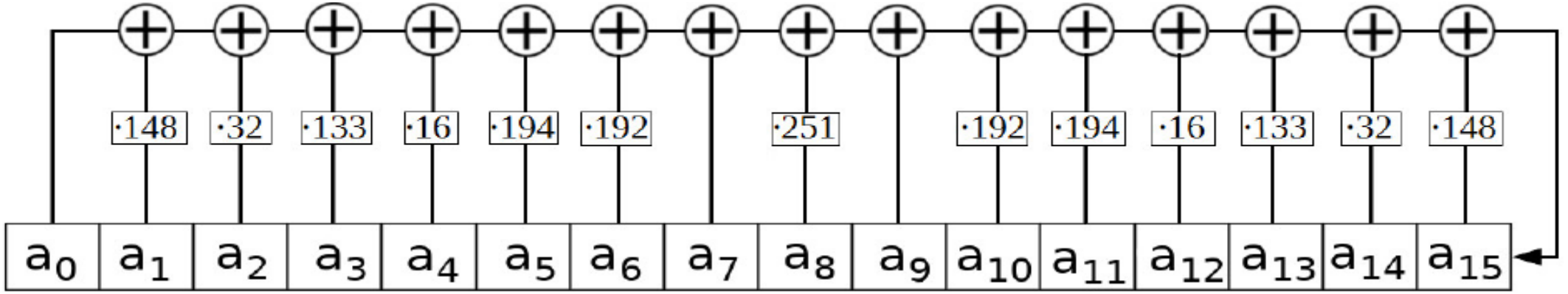
Согласно ГОСТ 34.11-2015 – Блочные шифры:

4.1.2 Линейное преобразование

Линейное преобразование задается отображением $\ell: V_8^{16} \to V_8$, которое определяется следующим образом:

$$\ell(a_{15}, \ldots, a_0) = \nabla(148 \cdot \Delta(a_{15}) + 32 \cdot \Delta(a_{14}) + 133 \cdot \Delta(a_{13}) + 16 \cdot \Delta(a_{12}) + 194 \cdot \Delta(a_{11}) + 192 \cdot \Delta(a_{10}) + 1 \cdot \Delta(a_9) + 251 \cdot \Delta(a_8) + 1 \cdot \Delta(a_7) + 192 \cdot \Delta(a_6) + 194 \cdot \Delta(a_5) + 16 \cdot \Delta(a_4) + 133 \cdot \Delta(a_3) + 32 \cdot \Delta(a_2) + 148 \cdot \Delta(a_1) + 1 \cdot \Delta(a_0)) \quad (1)$$

для любых $a_i \in V_8$, $i = 0, 1, \ldots, 15$, где операции сложения и умножения осуществляются в поле $\mathbb{F}$, а константы являются элементами поля в указанном ранее смысле.



Математическое описание умножение в поле Галуа над неприводимым многочленом α

Стандартный вариант	Предлагаемый вариант
$x = (x_m, x_{m-1}, ..., x_1, x_0);$	$x = (x_m, x_{m-1}, ..., x_1, x_0);$
$\alpha * x = \begin{cases} (x_{m-1}, ..., x_1, x_0, 0), & x_m = 0; \\ (x_{m-1}, ..., x_1, x_0, 0) \oplus \alpha, & x_m = 1. \end{cases}$	$\alpha * x = (x_{m-1}, ..., x_1, x_0, 0) \oplus [\alpha \cdot \{x_m\}].$
Подход применен в [6, 9, 10, 11, 12]	

Реализация умножения в поле Галуа в преобразовании $L_{Алг}$

Стандартный вариант

$$L:$$

$$r = \bigoplus_{i=1}^{16} mult(x^{(i)}, y^{(i)})$$

$$mult(x, y):$$

$$r = r \oplus x, \text{ если } y_j = 1$$

$$\alpha * x = \begin{cases} (x_{m-1}, ..., x_1, x_0, 0), & x_m = 0; \\ (x_{m-1}, ..., x_1, x_0, 0) \oplus \alpha, & x_m = 1. \end{cases}$$

$$j = \overline{1...8}$$

$$y = \{251,\ 1, 192, 194, 16, 133,\ 32, 148,$$
$$1, 148,\ 32, 133, 16, 194, 192,\ 1\}.$$

Подход применен в [6, 9, 10, 11, 12]

Предлагаемый вариант

$$L:$$

$$r = \bigoplus_{i=1}^{16} \bigoplus_{j=1}^{8} \left[\begin{array}{c} x^{(i)}, \text{ если } y^{(i)}{}_j = 1 \\ x^{(i)} = (x^{(i)} << 1) \oplus \alpha \cdot (x^{(i)} >> 7) \end{array} \right]$$

$$y = \{251,\ 1, 192, 194, 16, 133,\ 32, 148,$$
$$1, 148,\ 32, 133, 16, 194, 192,\ 1\}.$$

Реализация умножения в поле Галуа в преобразовании $L_{Алг}$

Стандартный вариант

$$L:$$

$$r = \bigoplus_{i=1}^{16} mult(x^{(i)}, y^{(i)})$$

$$mult(x, y):$$

$$r = r \oplus x, \text{ если } y_j = 1$$

$$\alpha * x = \begin{cases} (x_{m-1}, ..., x_1, x_0, 0), & x_m = 0; \\ (x_{m-1}, ..., x_1, x_0, 0) \oplus \alpha, & x_m = 1. \end{cases}$$

$$j = \overline{1...8}$$

$$y = \{251,\ 1,\ 192,\ 194,\ 16,\ 133,\ 32,\ 148,\ 1,\ 148,\ 32,\ 133,\ 16,\ 194,\ 192,\ 1\}.$$

Подход применен в [6, 9, 10, 11, 12]

Предлагаемый вариант (I изменение)

$$L:$$

$$r = \bigoplus_{i=1}^{16} \underbrace{\bigoplus_{j=1}^{8} \left[\begin{array}{c} x^{(i)}, \text{ если } y^{(i)}{}_j = 1 \\ x^{(i)} = (x^{(i)} << 1) \oplus \alpha \cdot (x^{(i)} >> 7) \end{array} \right]}_{m^*}$$

$$y = \{251,\ 1,\ 192,\ 194,\ 16,\ 133,\ 32,\ 148,\ 1,\ 148,\ 32,\ 133,\ 16,\ 194,\ 192,\ 1\}.$$

Реализация умножения в поле Галуа в преобразовании $L_{Алг}$

Разложим m^*. Заметим, что последнее выражение не влияет на результат.

Следовательно, можно упростить m^*.

$$y = \{251,\ 1,\ 192,\ 194,\ 16,\ 133,\ 32,\ 148,$$
$$1,\ 148,\ 32,\ 133,\ 16,\ 194,\ 192,\ 1\}.$$

m^*:

$$r = r \oplus x^{(i)} * (y^{(i)}\ \ \&\ \ 0x01)$$

$$x^{(i)} = (x^{(i)} << 1) \oplus \alpha \cdot (x^{(i)} >> 7)$$

$$r = r \oplus x^{(i)},\ \text{если}\ y^{(i)}{}_2 = 1$$

$$x^{(i)} = (x^{(i)} << 1) \oplus \alpha \cdot (x^{(i)} >> 7)$$

.....

$$r = r \oplus x^{(i)},\ \text{если}\ y^{(i)}{}_8 = 1$$

$$x^{(i)} = (x^{(i)} << 1) \oplus \alpha \cdot (x^{(i)} >> 7)$$

Реализация умножения в поле Галуа в преобразовании $L_{Алг}$

Полученное выражение назовем m^{**}.

m^{**}:

$$r = r \oplus x^{(i)} * (y^{(i)} \ \& \ 0x01)$$

$$x^{(i)} = (x^{(i)} << 1) \oplus \alpha \cdot (x^{(i)} >> 7)$$

$$r = r \oplus x^{(i)}, \text{ если } y^{(i)}_2 = 1$$

$$x^{(i)} = (x^{(i)} << 1) \oplus \alpha \cdot (x^{(i)} >> 7)$$

.....

$$r = r \oplus x^{(i)}, \text{ если } y^{(i)}_8 = 1$$

$$y = \{251, \ 1, 192, 194, 16, 133, \ 32, 148,$$
$$1, 148, \ 32, 133, 16, 194, 192, \ 1\}.$$

Реализация умножения в поле Галуа в преобразовании $L_{Алг}$

Стандартный вариант

$$L:$$

$$r = \bigoplus_{i=1}^{16} mult(x^{(i)}, y^{(i)})$$

$$mult(x, y):$$

$$r = r \oplus x, \text{ если } y_j = 1$$

$$\alpha * x = \begin{cases} (x_{m-1}, \ldots, x_1, x_0, 0), & x_m = 0; \\ (x_{m-1}, \ldots, x_1, x_0, 0) \oplus \alpha, & x_m = 1. \end{cases}$$

$$j = \overline{1 \ldots 8}$$

$$y = \{251, \quad 1, 192, 194, 16, 133, \quad 32, 148,$$
$$1, 148, \quad 32, 133, 16, 194, 192, \quad 1\}.$$

Подход применен в [6, 9, 10]

Предлагаемый вариант(II изменение)

$$L:$$

$$r = \bigoplus_{i=1}^{8} (m^{**}_i \oplus m^{**}_{i+8})$$

Реализация умножения в поле Галуа в преобразовании $L_{Алг}$

m^{**}_i:

$r = r \oplus x^{(i)} \cdot (y^{(i)} \ \& \ 0x01)$

$x^{(i)} = (x^{(i)} << 1) \oplus \alpha \cdot (x^{(i)} >> 7)$

$r = r \oplus x^{(i)}$, если $y^{(i)}_2 = 1$

$x^{(i)} = (x^{(i)} << 1) \oplus \alpha \cdot (x^{(i)} >> 7)$

.....

$r = r \oplus x^{(i)}$, если $y^{(i+8)}_8 = 1$

m^{**}_{i+8}:

$r = r \oplus x^{(i+8)} \cdot (y^{(i+8)} \ \& \ 0x01)$

$x^{(i+8)} = (x^{(i+8)} << 1) \oplus \alpha \cdot (x^{(i+8)} >> 7)$

$r = r \oplus x^{(i+8)}$, если $y^{(i+8)}_2 = 1$

$x^{(i+8)} = (x^{(i+8)} << 1) \oplus \alpha \cdot (x^{(i+8)} >> 7)$

.....

$r = r \oplus x^{(i+8)}$, если $y^{(i+8)}_8 = 1$

$y = \{251, \ 1, 192, 194, 16, 133, \ 32, 148,$

$1, 148, \ 32, 133, 16, 194, 192, \ 1\}.$

Реализация умножения в поле Галуа в преобразовании $L_{Алг}$

Применительно к тем значениям *y*, которые представлены в

ГОСТ 34.11-2015 – Блочные шифры.

$$y = \{251,\ 1,\ 192,\ 194,\ 16,\ 133,\ 32,\ 148,$$
$$1,\ 148,\ 32,\ 133,\ 16,\ 194,\ 192,\ 1\}.$$

$m{**}$:

$r = r \oplus x^{(i)} * (y^{(i)}\ \ \&\ \ 0x01)$

$x^{(i)} = (x^{(i)} << 1) \oplus \alpha \cdot (x^{(i)} >> 7)$

если $y^{(i)} > 1$

$r = r \oplus x^{(i)},$ если $y^{(i)}{}_2 = 1$

$x^{(i)} = (x^{(i)} << 1) \oplus \alpha \cdot (x^{(i)} >> 7)$

.....

$r = r \oplus x^{(i)},$ если $y^{(i)}{}_6 = 1$

если $y^{(i)} > 127$

$x^{(i)} = (x^{(i)} << 1) \oplus \alpha * (x^{(i)} >> 7)$

$r = r \oplus x^{(i)},$ если $y^{(i)}{}_7 = 1$

$r = (x^{(i)} <<\ 1) \oplus \alpha * (x^{(i)} >> 7)$

Программная реализация умножения в поле Галуа в преобразовании $L_{Алг}$

Стандартный вариант

```
byte mul_gf(byte x, byte y) {
   byte p = 0;
   for (byte i = 0; i < 8; i++)
   {
     if ((y & 1) != 0)
        p ^= x;
     byte hi_bit_set = (byte)(x & 0x80);
     x <<= 1;
     if (hi_bit_set != 0)
        x ^= 0xC3; // x^8+x^7 + x^6 + x + 1
     y >>= 1;
   }
   return p;
}
```

Предлагаемый вариант

```
byte mul_gf(byte x, byte y) {
   byte p = 0;
   for (byte i = 0; i < 8; i++)
   {
     p ^= x * (y & (1 << i));
     x = (x << 1) ^ (0xC3 * (x >> 7));
   }
   return p;
}
```

Подход применен в [6, 9, 10, 11, 12]

Программная реализация умножения в поле Галуа в преобразовании $L_{Алг}$

Стандартный вариант

```
byte mul_gf(byte x, byte y)
{
   byte p = 0;
   while( y )
   {
     if ((y & 1) != 0)
        p ^= x;
     x = (x << 1) ^ (x & 0x80 ? 0xC3 : 0);
     y >>= 1;
   }
   return p;
}
```

Подход применен в [2]

Предлагаемый вариант

```
byte mul_gf(byte x, byte y)
{
   byte p = 0;
   while( y )
   {

     p ^= x * (y & 1);
     x = (x << 1) ^ (0xC3 * (x >> 7));
     y >>= 1;
   }
   return p;
}
```

Программная реализация умножения в поле Галуа в преобразовании $L_{Алг}$

Стандартный вариант

```
BYTE ConversionL(ULONGLONG block[ 2 ])
   BYTE a[ 16 ];
   for (BYTE i = 0; i < 16; i++)
      if (i < 8)
        a[ i ] = block[ 0 ] >> (8 * i);
      else
        a[ i ] = block[ 1 ] >> (8 * (i % 8));

   BYTE val = mul_gf( a[ 7 ], 148 );
   val ^=  mul_gf( a[ 6 ],  32 );
   val ^=  mul_gf( a[ 0 ], 251 );
   ..............................
   val ^=  mul_gf( a[ 15 ],   1);
   val ^=  mul_gf( a[ 14 ], 192);
   ..............................
   val ^=  mul_gf( a[  8 ],   1);

   return val;
```

Подход применен в [2, 6, 9, 10, 11, 12]

Предлагаемый вариант (II изменение)

```
BYTE ConversionL(ULONGLONG block[ 2 ])
  BYTE res = 0;
  for (BYTE i = 0; i < 8; i += 1)
  {
    x = block[ 0 ] >> (i << 3);
    y = aucKoeff[ i ];
    res ^= x * (y & 1);
    x = (x << 1) ^ (0xC3 * ((x) >> 7));
    if (y & 0x02)
      res ^= x;
    x = (x << 1) ^ (0xC3 * ((x) >> 7));
    ..............................
    if (y & 0x40)
      res ^= x;
    x = (x << 1) ^ (0xC3 * ((x) >> 7));
    if (y & 0x80)
      res ^= x;

    x = block[ 1 ] >> (i << 3);
    y = aucKoeff[i + 8];
    ..............................
  }
  return res;
```

Условия исследования реализаций L

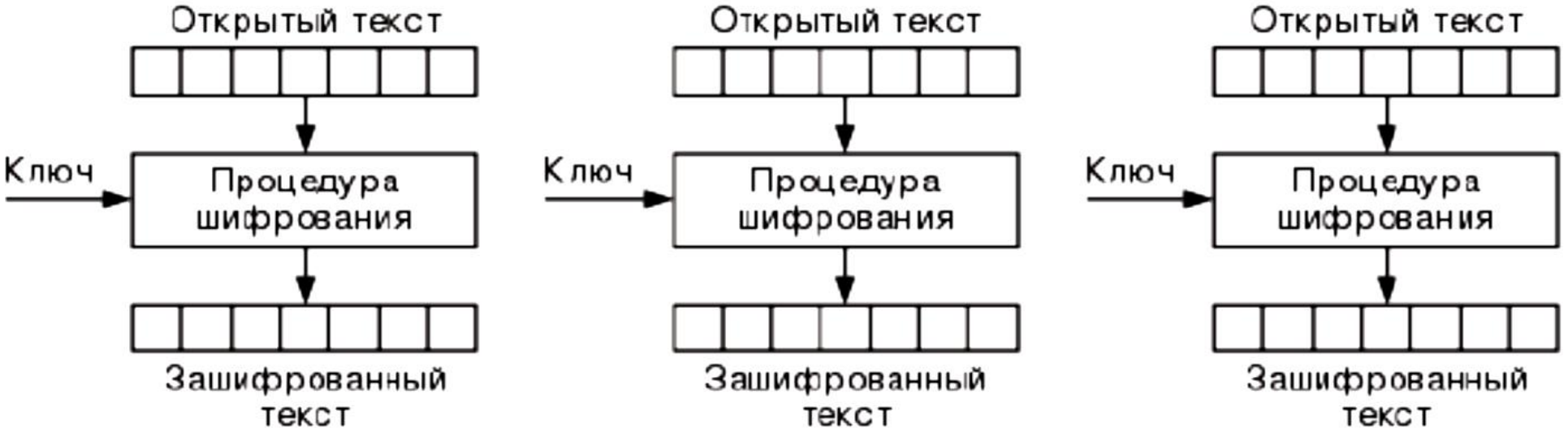
Введем обозначения и условия, на которых проведем исследование:

- $L_{Таб}$ – преобразование L с предвычисленной таблицей;
- $L_{Алг}$ – преобразование L, вычисляется алгоритмически;
- $L'_{Алг}$ – преобразование L, вычисляется алгоритмически предложенным методом;

Алгоритм	ГОСТ Р 34.12 2015
Режим шифрования	*ECB*
Входной блок	128 бит
Ключ K	32 байта
Таблица S	256 байт
Таблица L	65 536 байт
Количество потоков шифрования	1

Режим *ECB*

- *ECB* – режим электронной кодовой книги.
- Каждый блок шифруется независимо от других блоков.
- Шифрование: $C_i = E_k\ (P_i, k)$
- Дешифрование: $P_i = D_k(C_i, k)$



Экспериментальное исследование на различных вычислительных системах

Характеристики $L_{Алг}$, $L'_{Алг}$, $L_{Таб}$ были исследованы на:

- ЭВМ *Intel® Core™ i7-8700K CPU @***3.70 *GHz*** / 8 *Tb*
- Микроконтроллер *STM*32 *F*103 *C8T6*, **72 МГц**, отсутствует конвейер. Далее *F103*.
- Микроконтроллер *МИЛАНДР 1986ВЕ92У*, **80 МГц**, отсутствует конвейер. Далее *92У*.
- Микроконтроллер *ELVEES 1892ВМ15Ф*, **120 МГц**. Далее *15Ф*.

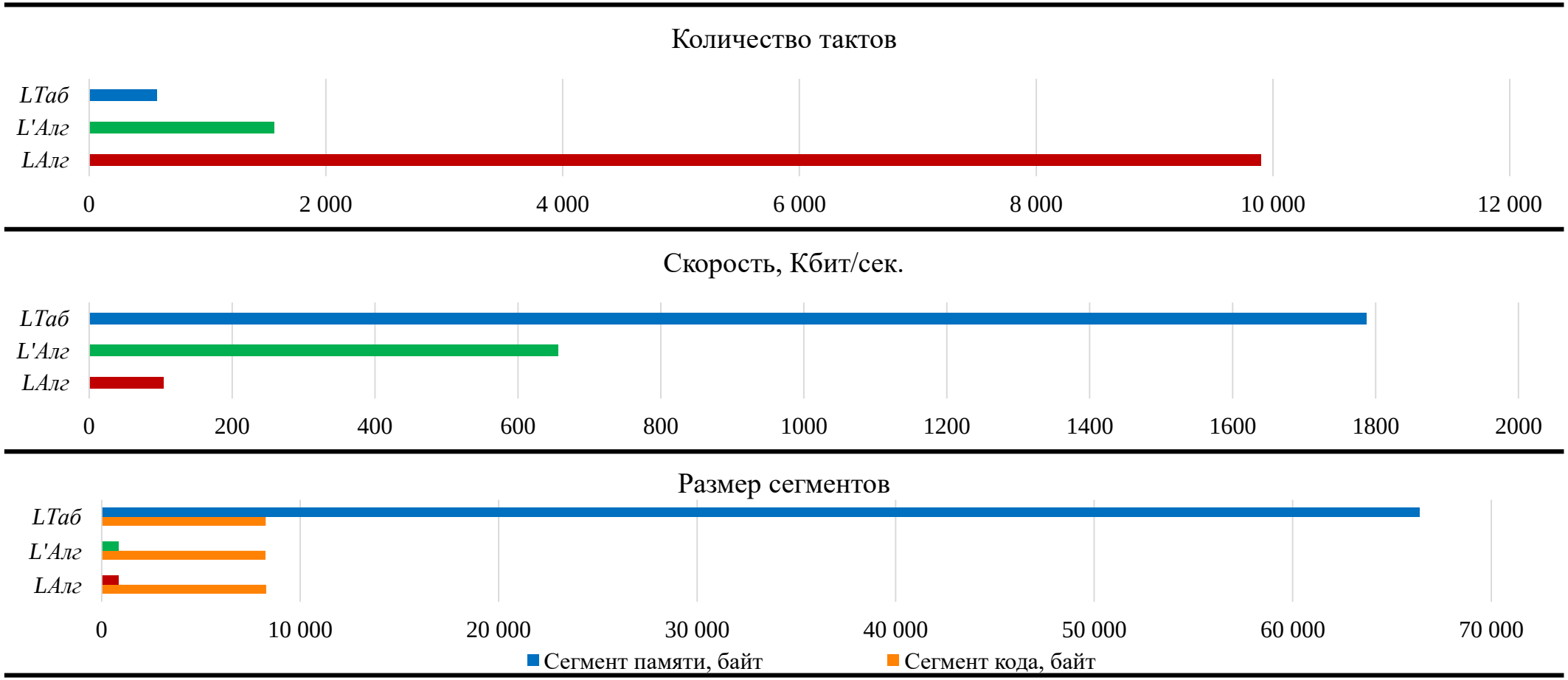
Результаты вычислительного эксперимента

- ЭВМ *InWin MidiTower ATX* / *Power Intel® Core™ i7-8700K CPU @ 3.70 GHz* / 8 *Tb*

Данные, 1 МБайт	$L_{Алг}$	$L'_{Алг}$	$L_{Таб}$
Такты, *getTickCount()*	**9 894**	**1014**	**573**
Время, сек	9,894	1,014	0,573
Скорость, Кбайт/сек.	103,49	1009,86	1787,08
Скорость, Кбит/сек.	**827,97**	**8078,89**	**14296,68**
Сегмент кода, байт	8 289	8 259	8 272
Сегмент памяти, байт	**835**	**851**	**66 371**

Результаты

- ЭВМ *InWin MidiTower ATX / Power Intel® Core™ i7-8700K CPU @ 3.70 GHz / 8 Tb.*



Результаты

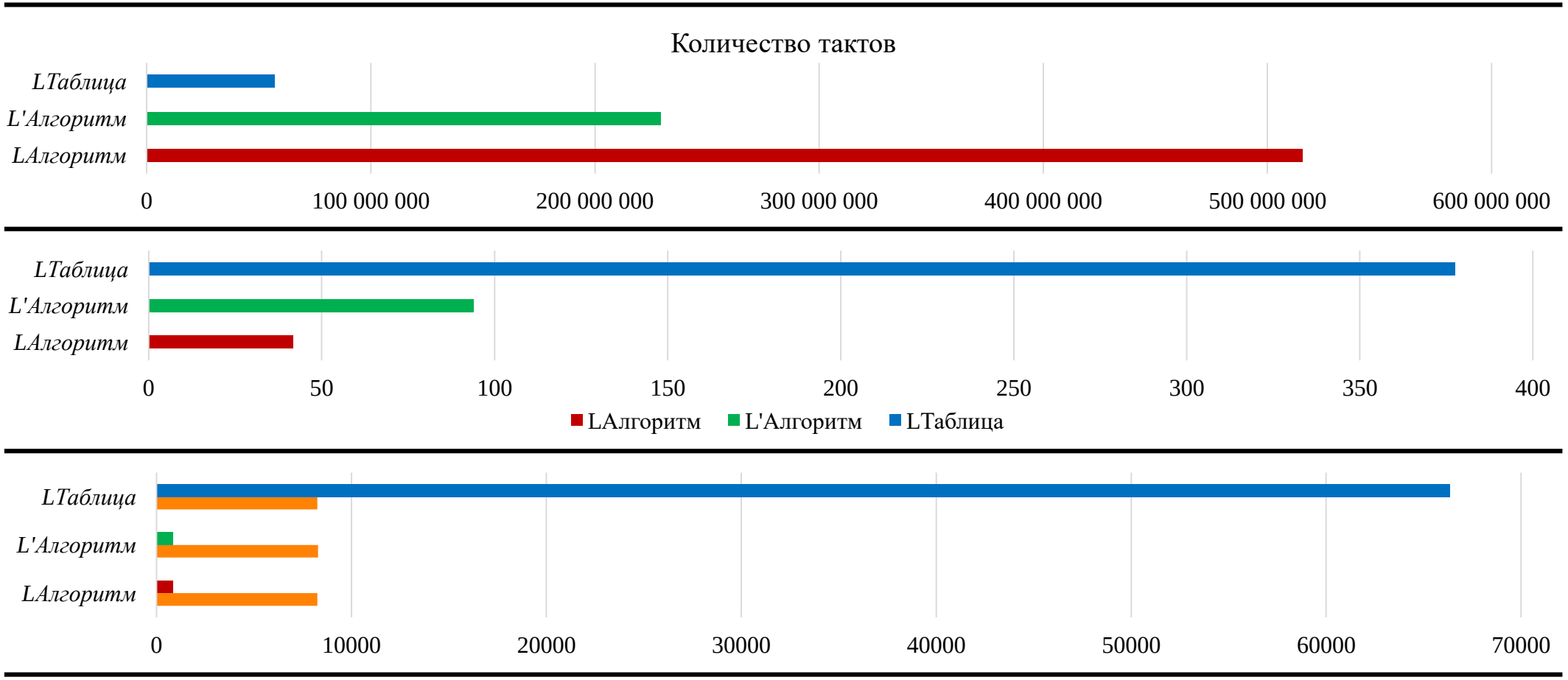
- Микроконтроллер *STM32 F407 VGT6*, 168 МГц

Данные, 1 Кбайт	$L_{Алг}$	$L'_{Алг}$	$L_{Таб}$
Такты, *DWT*	**515 658 771**	**229 309 463**	**56 953 880**
Время, сек.	3,07	1,36	0,34
Скорость, Кбайт/сек.	5,21	11,72	47,20
Скорость, Кбит/сек.	**41,70**	**93,77**	**377,57**
Сегмент кода, байт	8245	8289	8245
Сегмент памяти, байт	**835**	**851**	**66371**

Результаты

- Микроконтроллер *STM32 F407 VGT6*, 168 МГц

Сравнение количества тактов, скорости и потребления памяти.



Результаты

- Микроконтроллер *STM*32 *F103 C8T6*, 72 МГц, Flash/RAM 64K/20K

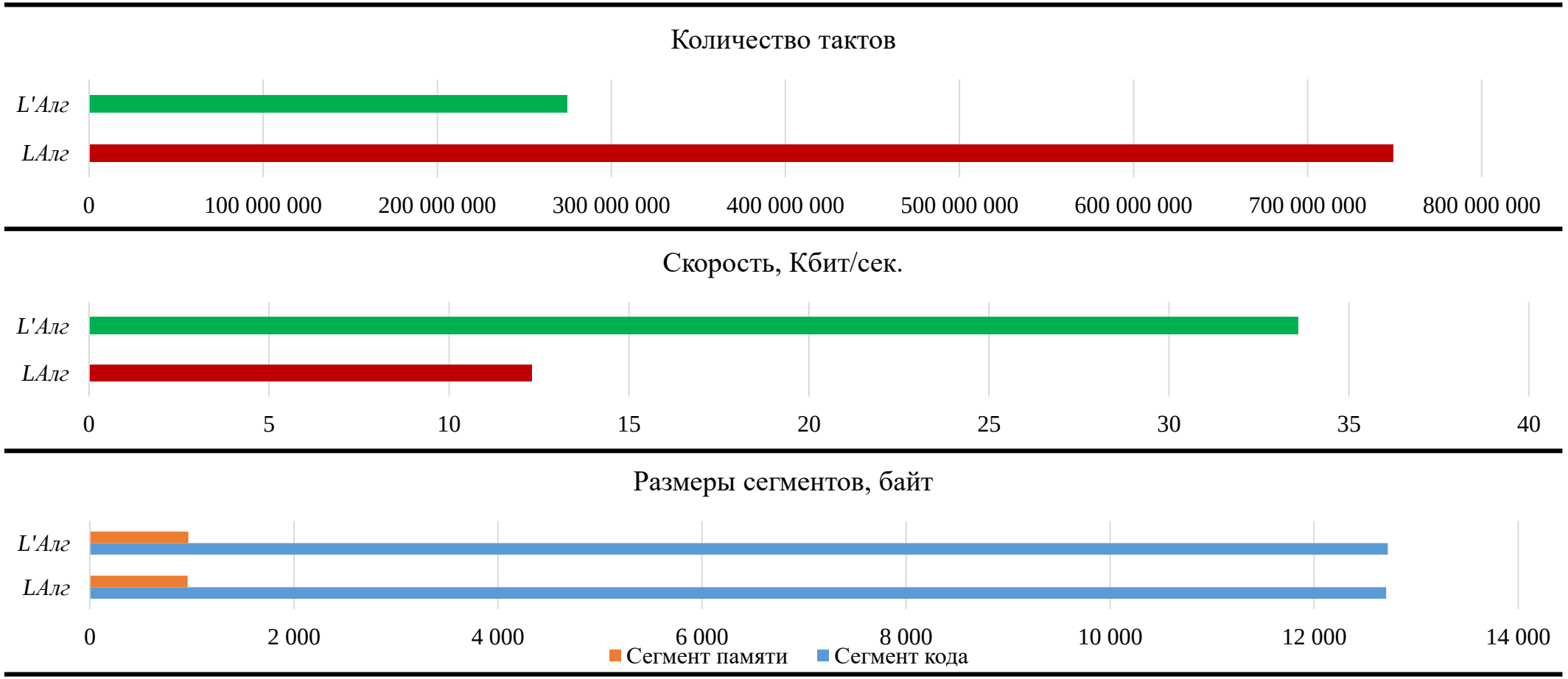
Данные, 16 КБайт	$L_{Алг}$	$L'_{Алг}$
Такты, *DWT*	**749 019 163**	**274 434 073**
Время, сек.	10,40	3,81
Скорость, Кбайт/сек.	1,53	4,19
Скорость, Кбит/сек.	**12,30**	**33,58**
Сегмент кода, байт	12 707	12 723
Сегмент памяти, байт	957	965

Технические характеристики не позволяют использовать $L_{Таб}$

Результаты

- Микроконтроллер *STM32 F103 C8T6*, 72 МГц, Flash/RAM 64K/20K

Сравнение количества тактов, скорости и потребления памяти.



Результаты

- Микроконтроллер *STM*32 *F103 C8T6*, 72 МГц, *Flash/RAM* 64K/20K

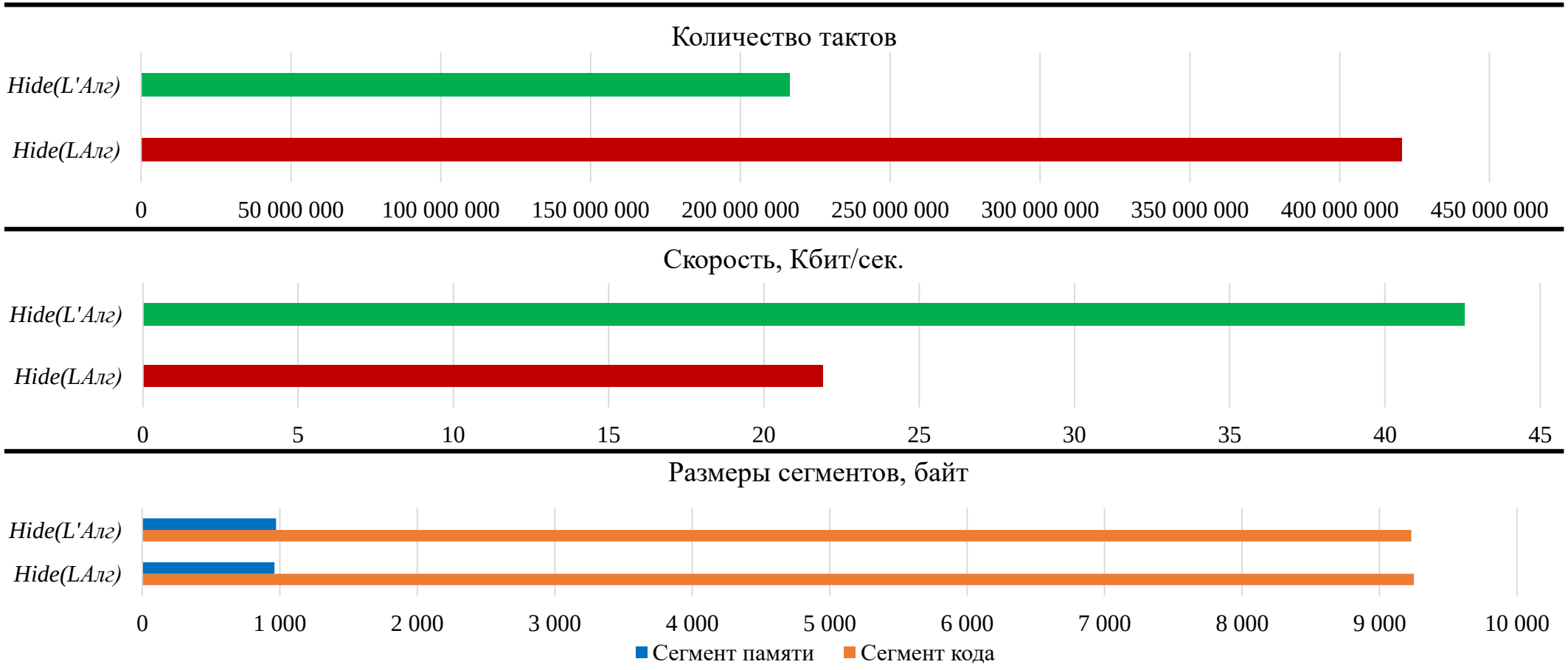
Данные, 16 КБайт	*Hide*($L_{Алг}$)	*Hide*($L'_{Алг}$)
Такты, *DWT*	**420 724 737**	**216 525 826**
Время, сек.	5,84	3,00
Скорость, Кбайт/сек.	2,73	5,32
Скорость, Кбит/сек.	**21,90**	**42,56**
Сегмент кода, байт	9 251	9 231
Сегмент памяти, байт	961	973

*Применим сбалансированную оптимизацию компилятора *IAR Embedded Workbench IDE*.

Результаты

- Микроконтроллер *STM32 F103 C8T6*, 72 МГц, *Flash/RAM* 64K/20K

Сравнение количества тактов, скорости и потребления памяти.



*Применим сбалансированную оптимизацию компилятора *IAR Embedded Workbench IDE*.

Результаты

- Микроконтроллер *МИЛАНДР 1986ВЕ92У*, 80 МГц, *Flash/RAM* 128К/32К

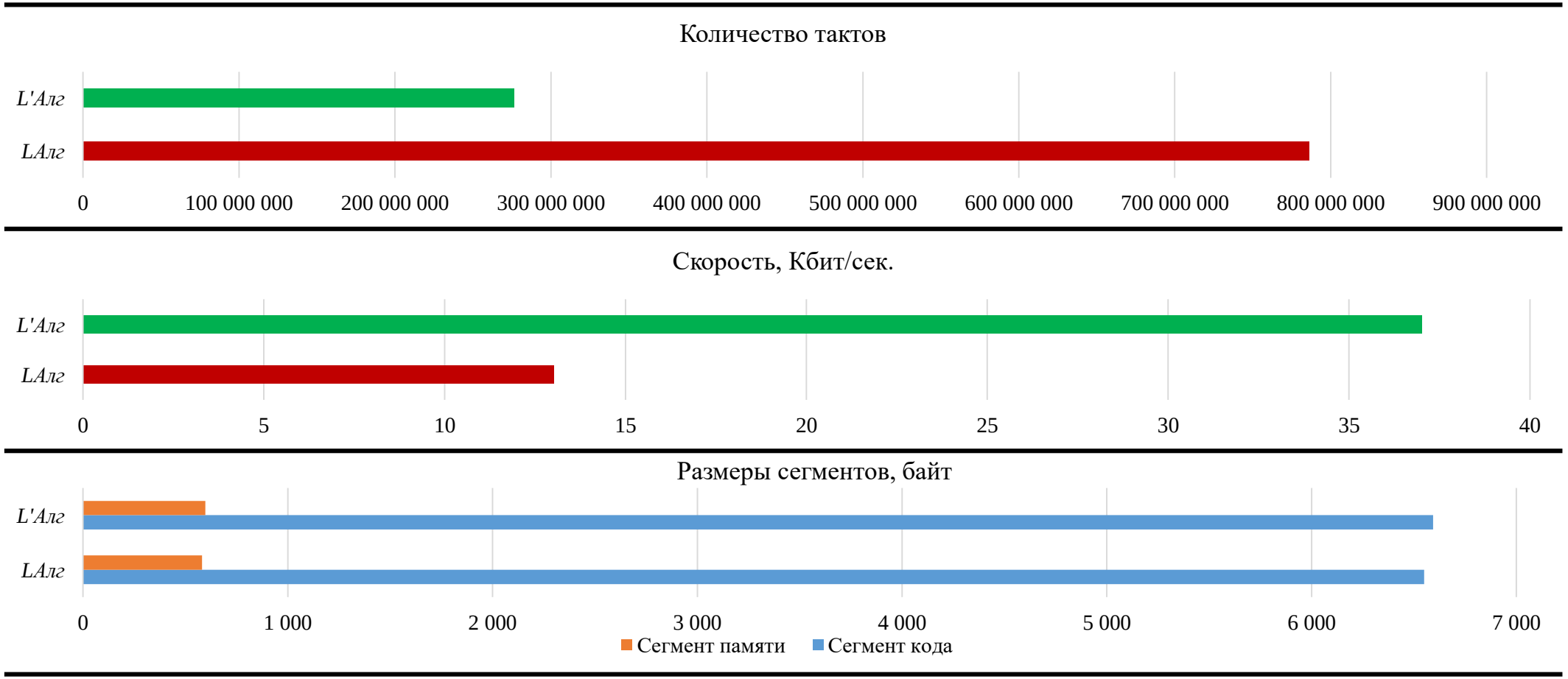
Данные, 16 КБайт	$L_{Алг}$	$L'_{Алг}$
Такты, *DWT*	**786 090 492**	**276 595 715**
Время, сек.	9,82	3,45
Скорость, Кбайт/сек.	1,62	4,62
Скорость, Кбит/сек.	**13,02**	**37,02**
Сегмент кода, байт	6 549	6 593
Сегмент памяти, байт	581	597

Технические характеристики не позволяют использовать $L_{Таб}$

Результаты

- Микроконтроллер *МИЛАНДР 1986ВЕ92У*, 80 МГц, *Flash/RAM* 128К/32К

Сравнение количества тактов, скорости и потребления памяти.



Результаты

- Микроконтроллер *МИЛАНДР 1986ВЕ92У*, 80 МГц, *Flash/RAM* 128К/32К

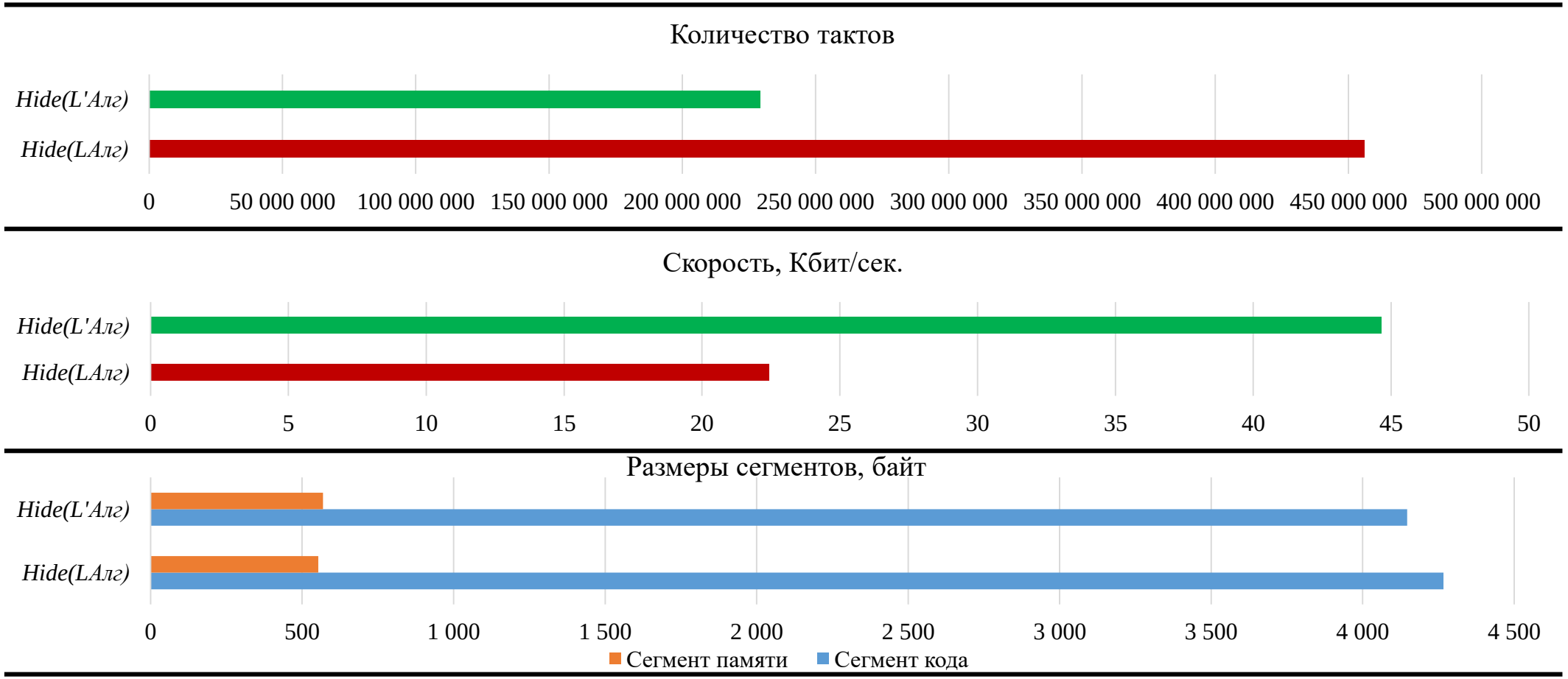
Данные, 16 КБайт	*Hide*($L_{Алг}$)	*Hide*($L'_{Алг}$)
Такты, *DWT*	**456 046 411**	**229 310 464**
Время, сек.	5,70	2,86
Скорость, Кбайт/сек.	2,80	5,58
Скорость, Кбит/сек.	**22,45**	**44,65**
Сегмент кода, байт	5 171	4 147
Сегмент памяти, байт	581	569

*Применим сбалансированную оптимизацию компилятора *IAR Embedded Workbench IDE*.

Результаты

- Микроконтроллер *МИЛАНДР 1986ВЕ92У*, 80 МГц, *Flash/RAM* 128К/32К

Сравнение количества тактов, скорости и потребления памяти.



*Применим сбалансированную оптимизацию компилятора *IAR Embedded Workbench IDE*.

Результаты

- Микроконтроллер *ELVEES 1892ВМ15Ф, 120 МГц.*

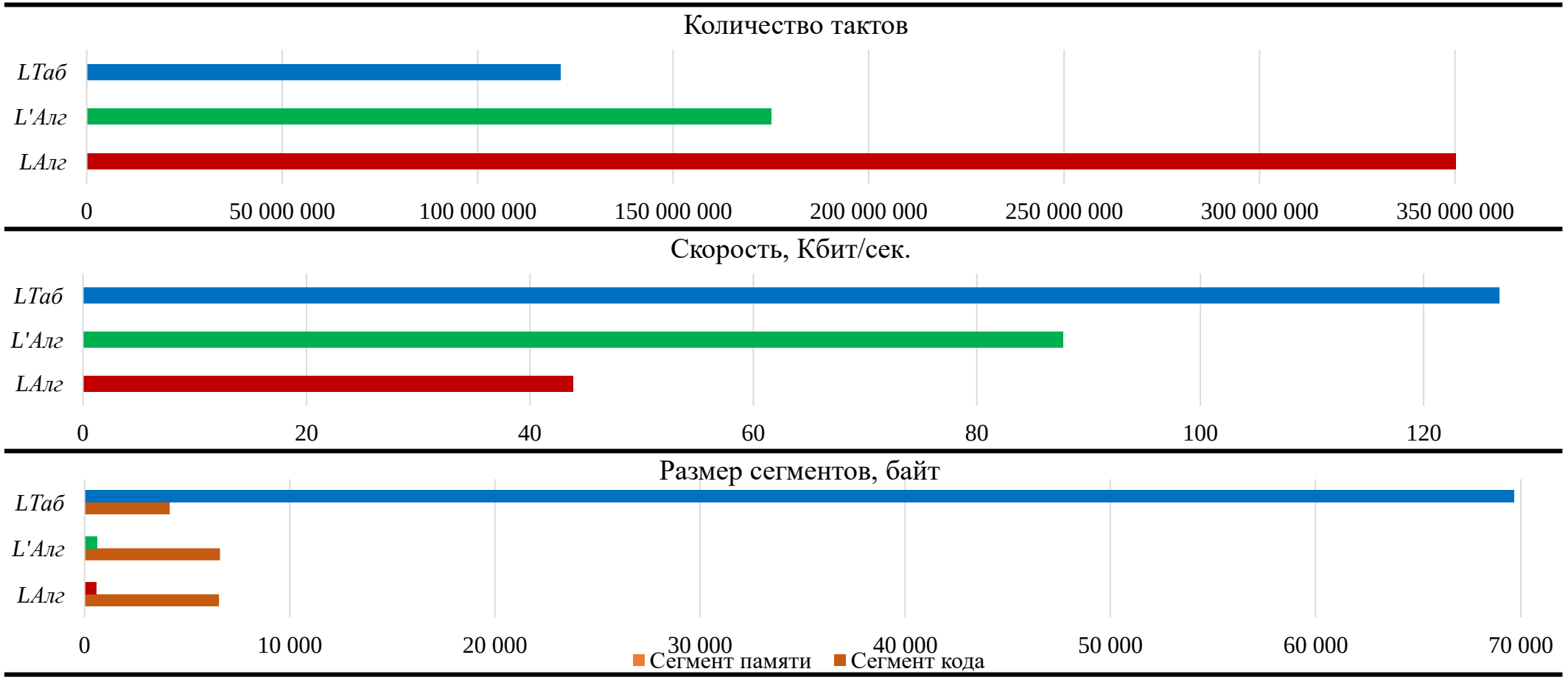
Данные, 16 КБайт	$L_{Алг}$	$L'_{Алг}$	$L_{Таб}$
Такты	**350 219 578**	**175 109 789**	**121 146 388**
Время, сек.	2,918496483	1,459248242	1,009553233
Скорость, Кбайт/сек.	5,482274894	10,96454979	15,84859468
Скорость, Кбит/сек.	**43,85819916**	**87,71639831**	**126,7887574**
Сегмент кода	6 549	6 593	4 147
Сегмент памяти	**581**	**597**	**69 683**

*Применим оптимизацию -O2 компилятора MS Studio 4 (2019.07.175)

Результаты

- Микроконтроллер *ELVEES 1892ВМ15Ф, 120 МГц.*

Сравнение количества тактов, скорости и потребления памяти.



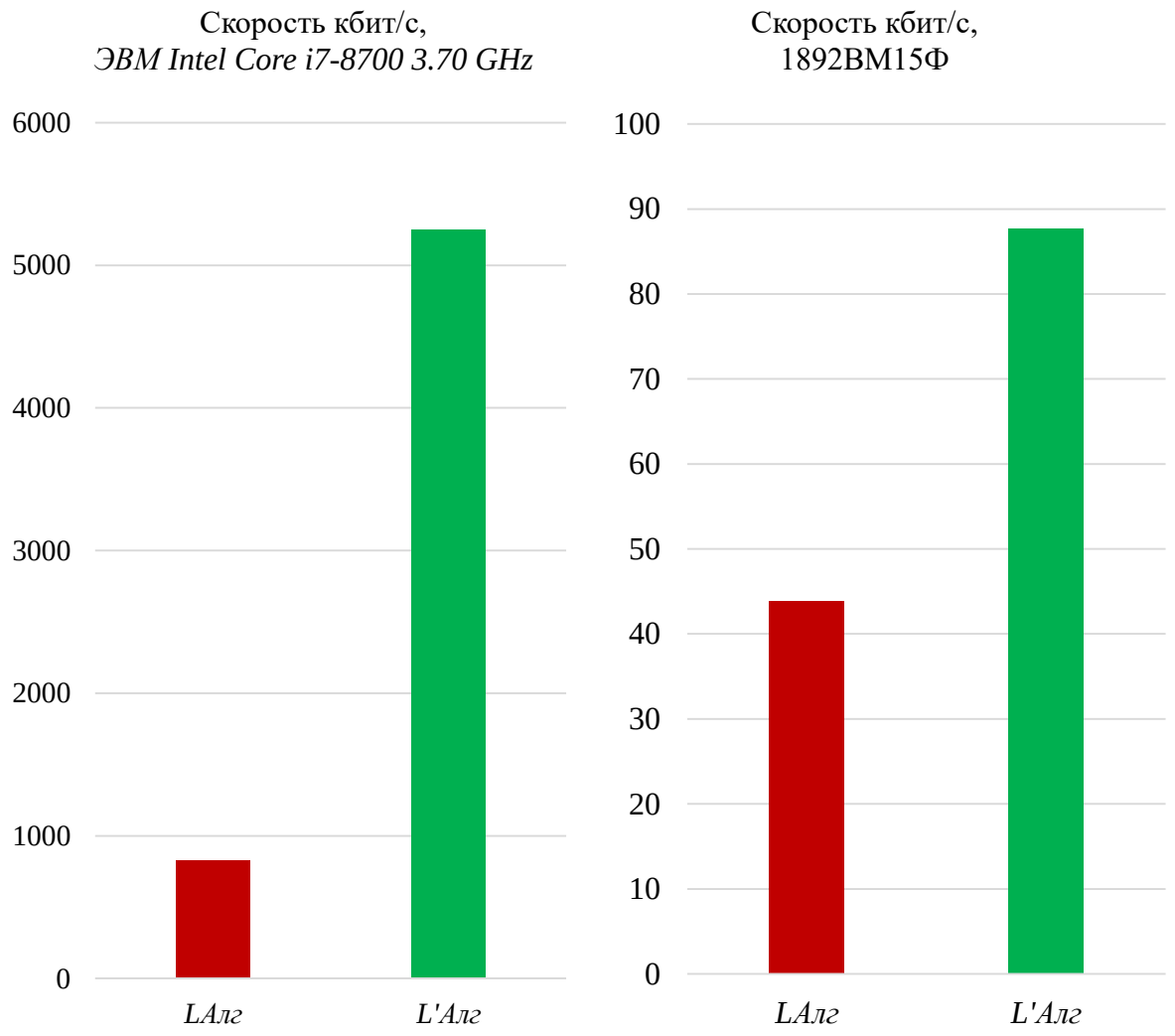
*Применим оптимизацию -O2 компилятора MS Studio 4 (2019.07.175)

Заключение

- Предложен **экономичный** вариант алгоритмического **умножения в поле Галуа** в преобразования *L*.

- **Не требуется** хранить таблицу предвычислений 64 Кбайт.

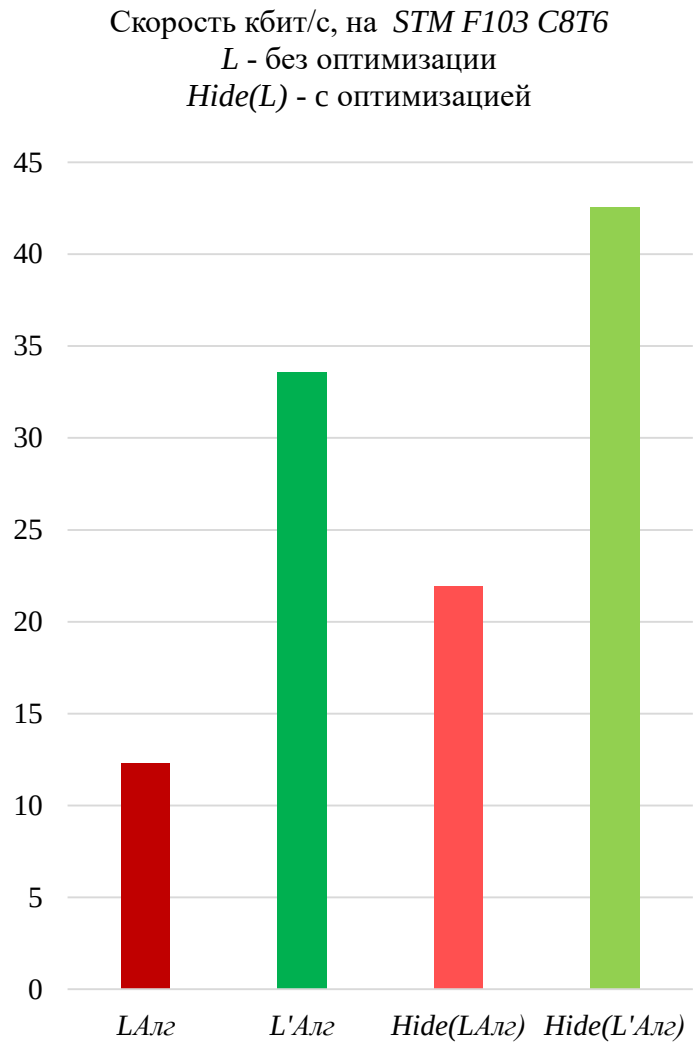
Тем самым, объём сегмента памяти **уменьшен в 78 раз**.

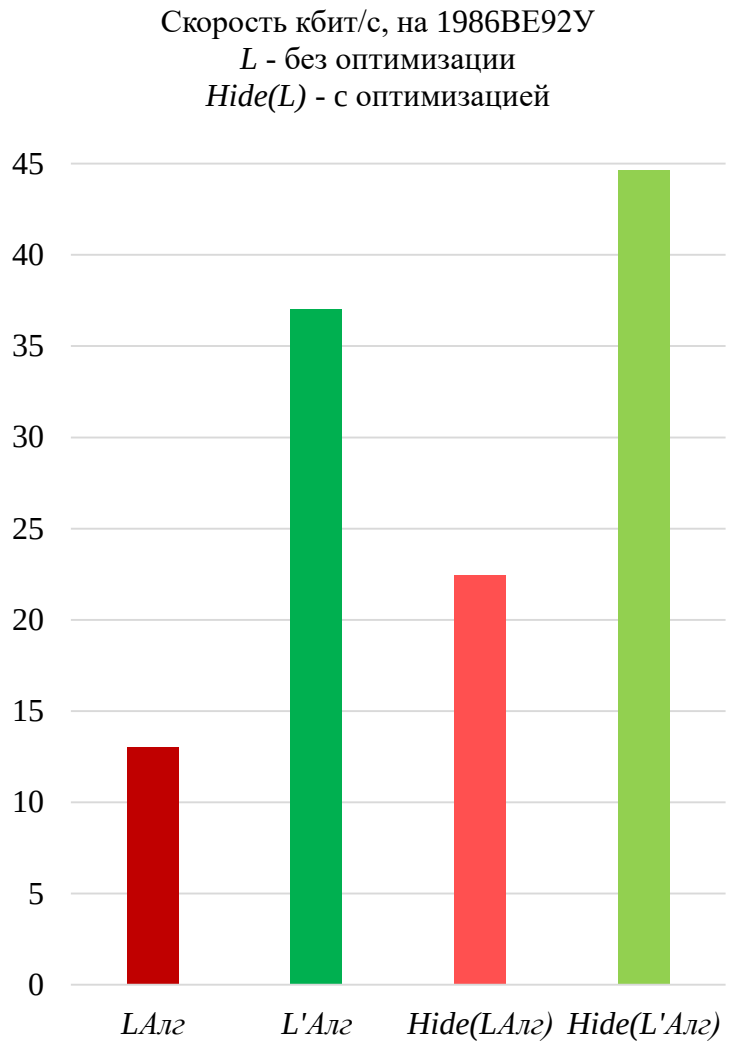


Заключение

- Предложенный вариант **реализуем** на микроконтроллерах с ограниченными ресурсами;

- Предложенный вариант **независим** от введения **нового** полинома и коэффициентов умножения.





Заключение

- Предложенный вариант **для ЭВМ** L'_A по сравнению с L_A , позволяет достичь **экономии времени на 90%**, т. е. L'_A быстрее L_A **в 10 раз**.
- Предложенный вариант **для микроконтроллеров**: L'_A быстрее L_A **в 2-3 раза**.
- **На графических** процессорах L'_A не требует взаимодействия с памятью. Таким образом применив L'_A можно получить результат не ниже существующего.
- Полученные в ходе исследования результаты применимы **в других** вычислительных методах, использующие умножение в поле Галуа.
- Метод применим **для большей степени** неприводимого многочлена.

Благодарю за внимание!