

**Исследование вычислительных
свойств ассоциативного
решающего поля на примере задач
сортировки массивов**

Руководитель работы

доцент, к.т.н.

Кафтанников И.Л.

Автор работы

Студентка группы КЭ-222

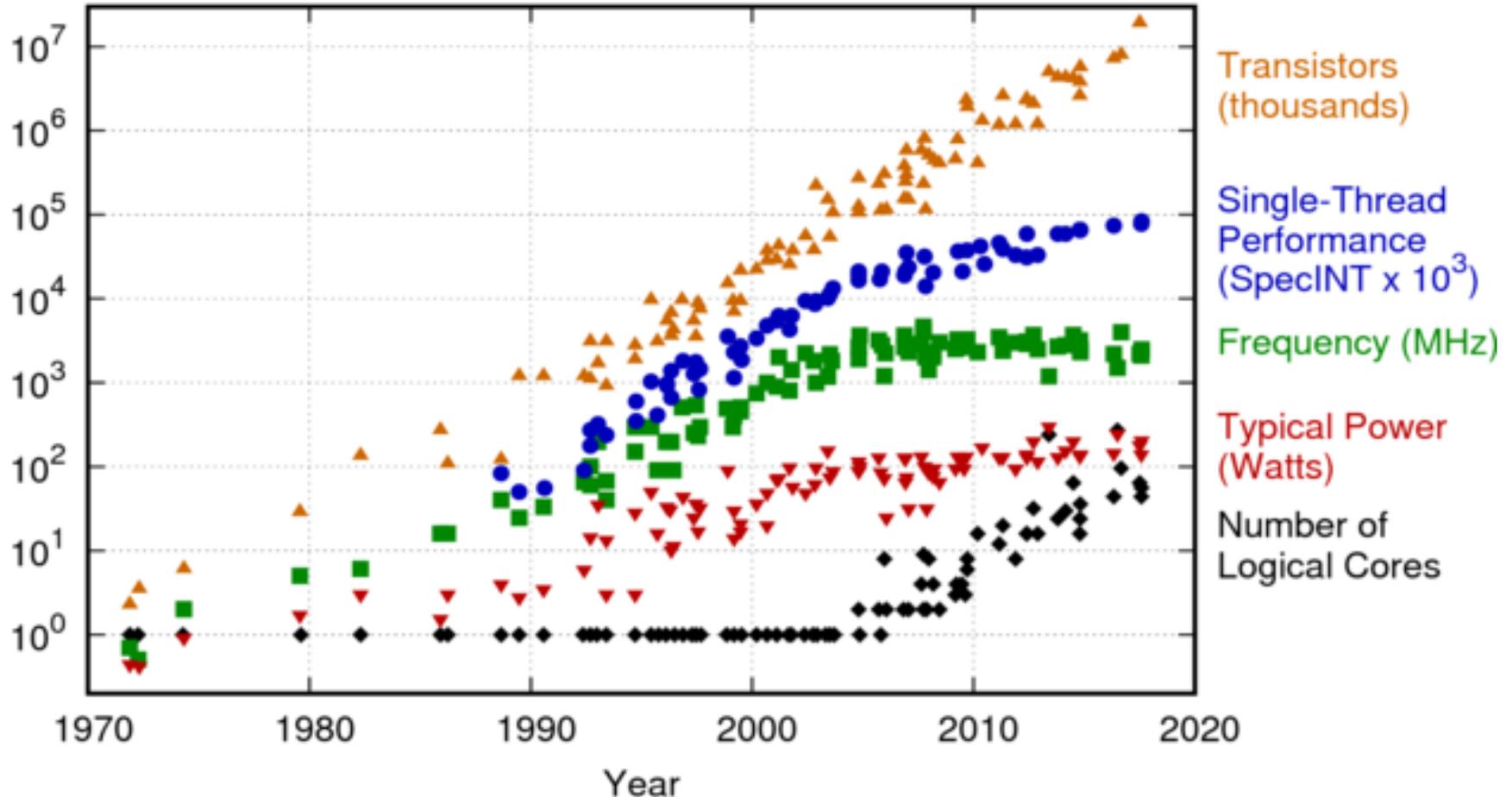
Прасолова А. Д.

Оглавление

1. [Увеличение числа транзисторов на кристалле микросхемы](#)
2. [Compute-centric Model и Data-centric Model](#)
3. [Отображение алгоритма на систему команд](#)
4. [Массив ячеек MRAM](#)
5. [Цель](#)
6. [Задачи](#)
7. [Структура ассоциативного решающего поля](#)
8. [Математическое описание операций сравнения элементов массива с константой](#)
9. [Последовательная и Послайсная загрузка данных](#)
10. [Понятие «сортировка»](#)
11. [Отображение классических алгоритмов сортировки на АП](#)
12. [Алгоритм нахождения максимального элемента в массиве](#)

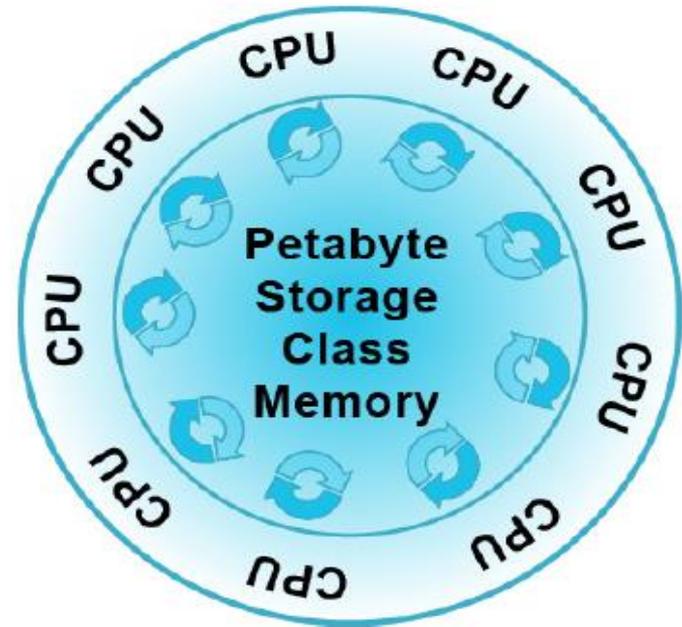
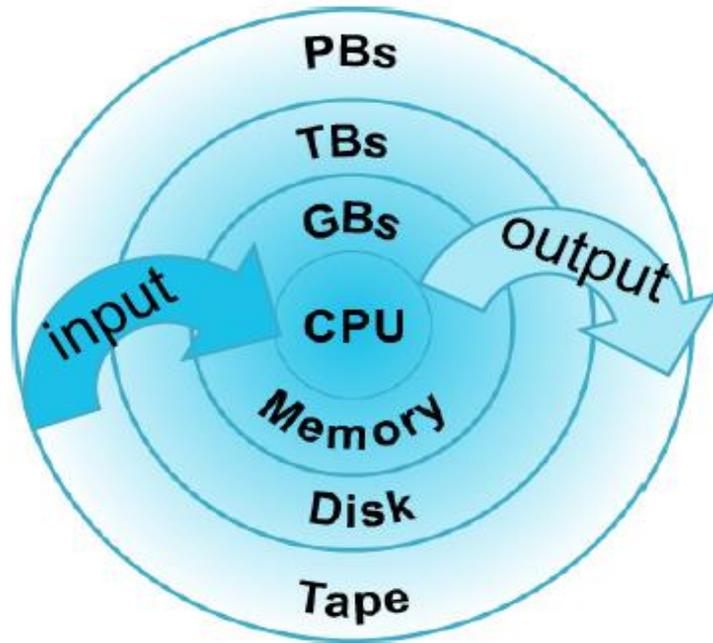
Увеличение числа транзисторов на кристалле микросхемы

42 Years of Microprocessor Trend Data



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp

Compute-centric Model и Data-centric Model



Отображение алгоритма на систему КОМАНД

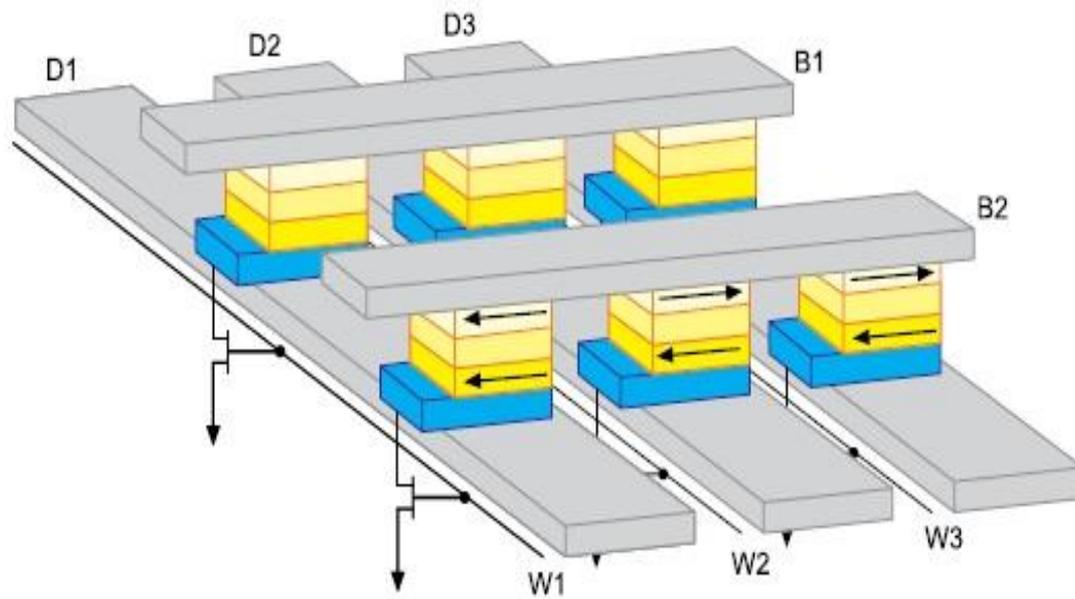
$$z_i = \{a_j^z\}$$

$$a_j = \langle \{C_k^j\}, \{T_k^j\} \rangle$$

$$a_j = \{O_l^{\text{ЯВУ}}\}$$

$$O_l = \{k_{CPU}^l\}$$

Массив ячеек MRAM



Цель

Цель работы – провести теоретические исследования по возможности реализации в АРП различных известных алгоритмов сортировки и разработать алгоритм, повышающий производительность с учетом структуры и свойств АРП.

Задачи

Для достижения поставленной цели необходимо решить следующие задачи:

- исследование и выявление особенностей функционирования ассоциативного решающего поля;
- анализ существующих разработок АРП;
- анализ и исследование отображения существующих алгоритмов сортировки на АРП;
- разработка нового алгоритма сортировки, учитывая особенности архитектуры и функционирования АП.

Структура ассоциативного решающего поля

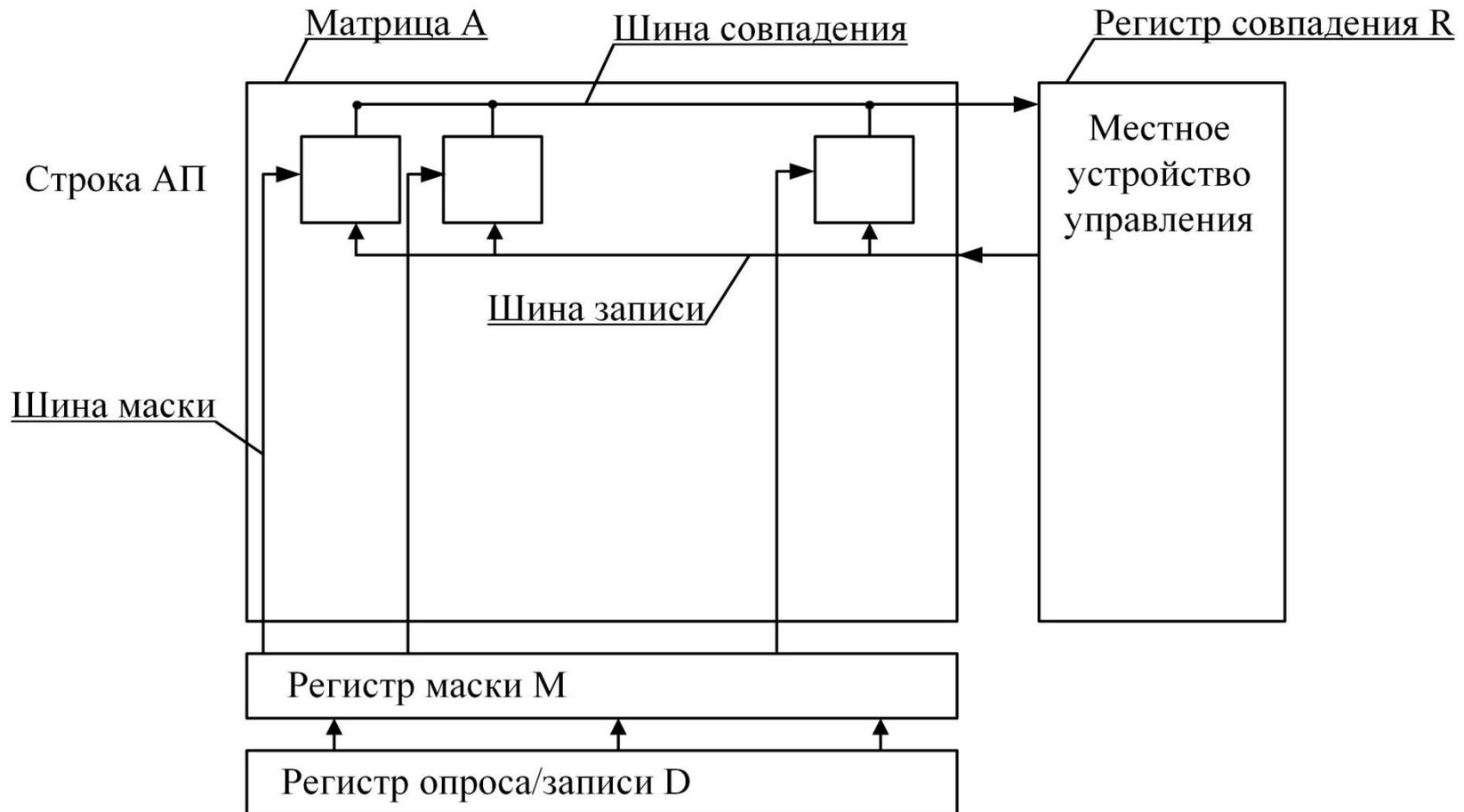
$$A_{NL} = \left\{ \begin{array}{ccc} a_{11} & \cdots & a_{1L} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & a_{NL} \end{array} \right\}$$

$$D = \langle d_1, d_2, \dots, d_L \rangle$$

$$M = \langle m_1, m_2, \dots, m_L \rangle$$

$$R = \langle r_1, r_2, \dots, r_N \rangle$$

Структура ассоциативного решающего поля



Математическое описание операций сравнения элементов массива с константой

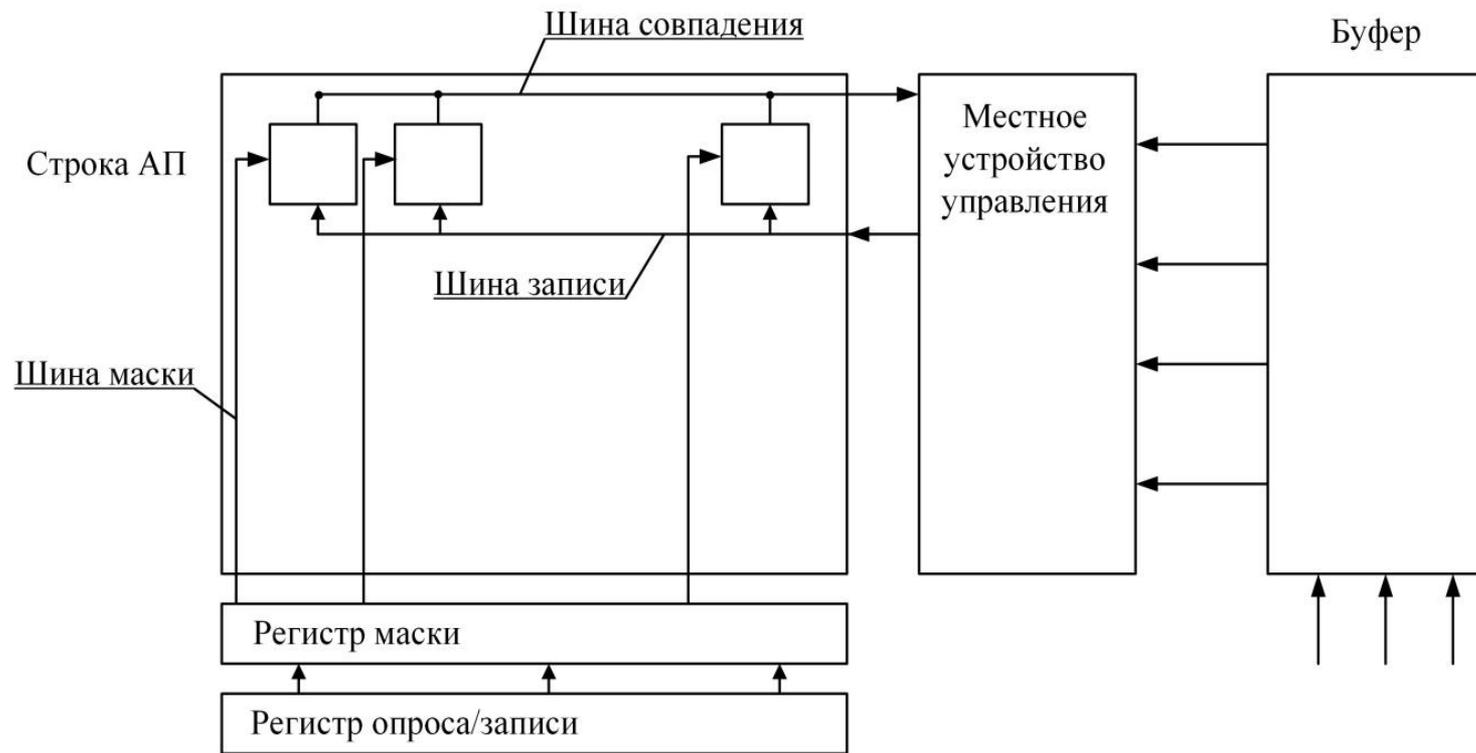
$$f_{\text{яч}_l}^{\text{опр}} = (a_{nl} \equiv d_l \wedge m_l)$$

$$f_{\text{стр}_n}^{\text{опр}} = \bigwedge_{l=1}^L f_{\text{яч}_l}^{\text{опр}} = f_{\text{яч}_1}^{\text{опр}} \wedge f_{\text{яч}_2}^{\text{опр}} \wedge \dots \wedge f_{\text{яч}_L}^{\text{опр}}$$

$$f_{\text{совп}} = \bigvee_{n=1}^N r_n = r_1 \vee r_2 \vee \dots \vee r_N$$

$$f_{\text{совп}} = \begin{cases} 1, & \text{если } \exists r_n \in R, r_n = 1 \\ 0, & \text{если } \forall r_n \in R, r_n = 0 \end{cases}$$

Последовательная и Послайсная загрузка данных



$$1. \exists m_l \in M; r_n \in R, (m_l \wedge r_n = 1) \Rightarrow a_{nl} := d_l | l = \overline{1, L}, n = const$$

$$2. \exists m_l \in M, (m_l \wedge b_{n1} = 1) \Rightarrow a_{nl} := b_{n1} | l = const$$

Понятие «сортировка»

Набор значений $k_i = \text{Random}()$

Адреса $a_i < a_{i+1}; a_{i+1} = a_i + 1$

Исходный массив $M = \{(k_i, a_i)\}$

$S = f_{\text{сорт}}(k) \left| \begin{array}{l} a_i > a_j \Rightarrow k_i > k_j - \text{сорт. по возр.} \\ a_i < a_j \Rightarrow k_i > k_j - \text{сорт. по убыв.} \end{array} \right.$

Отображение классических алгоритмов сортировки на АП

	Классическая временная сложность (в среднем)	Временная сложность для АП
Пузырьковая сортировка	$O(n^2)$	$O(n * t)$
Сортировка вставками	$O(n^2)$	$O(n * t)$
Сортировка выбором	$O(n^2)$	$O(n * t)$
Бисерная сортировка	—	$O(2^n)$

n – количество элементов в массиве;

t – разрядность.

Алгоритм нахождения максимального элемента в массиве

1. $k = 1$.
2. $d_k = 1; m_k = 1; m_j = 0; j = \overline{k + 1, L}$
3. *Опросить все доступные элементы на совпадение ($z_i \neq 1$).*
4. $\exists i = \overline{1, N} (r_i = 1 \wedge z_i = 0) \Rightarrow k + +; \text{п. 2}$
5. $\forall i = \overline{1, N} (r_i = 0 \wedge z_i = 0) \Rightarrow$
 $d_k = 0; m_k = 1; m_j = 0; j = \overline{k + 1, L}; \text{п3}$

Алгоритм нахождения максимального элемента в массиве

№ строки	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	CP
1	0	1	1	0	0	1	0	0	1	1	1	0	1	0	1	1	
2	0	1	0	0	1	1	1	0	1	0	0	1	1	1	0	0	
3	0	0	1	1	0	0	0	0	0	1	1	0	0	0	0	1	
4	1	0	1	1	1	0	1	1	0	1	0	1	0	1	0	1	
5	1	1	1	0	0	0	1	1	1	1	0	0	0	1	0	1	
6	0	0	0	1	1	0	1	0	0	0	1	1	0	1	1	0	
7	1	0	1	1	1	1	0	1	0	1	1	1	1	0	0	0	
8	1	1	0	0	0	0	0	1	1	0	0	0	0	0	1	0	
	⋮								⋮							⋮	
N-5	0	0	1	0	1	1	0	0	0	1	1	1	1	0	1	1	
N-4	0	1	1	0	0	1	1	0	1	1	0	0	1	1	1	0	
N-3	1	1	1	1	0	1	0	0	0	0	0	0	1	0	0	0	
N-2	0	1	0	1	1	0	0	0	1	0	1	1	0	0	1	1	
N-1	1	1	1	1	0	0	1	1	0	1	0	1	1	1	0	1	
N	1	0	0	1	0	0	1	1	0	0	1	0	0	1	1	0	

Спасибо за внимание!