Сложность базовых булевых операторов для цифровой схемотехники

И. С. Сергеев*

Статья содержит обзор оценок сложности схем для базовых булевых преобразований, применяемых в цифровой схемотехнике, и эффективных методов синтеза таких схем. Изложение охватывает структурно простые функции и операторы, такие как счетчики, сумматоры, шифраторы, мультиплексоры, и исключает более сложные алгебраические операции с числами, многочленами и матрицами. Дополнительно рассмотрено несколько приложений к построению схем более узкого назначения.

Ключевые слова: булевы схемы, сложность, глубина, параллельные схемы, префиксные схемы, инкрементор, двусторонний счетчик, счетчик Грея, сумматор, компаратор, дешифратор, мультиплексор, шифратор, компрессор, пороговые симметрические функции, приоритетный шифратор, унарная кодировка, сортировка.

Введение

В работе предпринимается попытка собрать сведения о сложности самых простых и фундаментальных булевых преобразований, которые широко используются как в практической схемотехнике, так и в теоретических задачах синтеза схем. Речь идёт о шифраторах, мультиплексорах, компараторах и прочих операторах, структурно не более сложных, чем оператор сложения чисел. Рассматриваемые преобразования часто не имеют выраженного самостоятельного значения, а играют роль «строительных блоков» при решении содержательных задач. С этих преобразований обычно начинается изучение основ проектирования цифровых функциональных узлов в популярных учебниках по цифровой схемотехнике, см., например, [9, 11]. Поэтому в рамках настоящего обзора мы (неформально) называем их *базовыми*.

Теория алгоритмов умножения чисел или матриц, дискретных преобразований Фурье настолько развита, что требует для изложения многотомных изданий. При этом «рабочие лошадки» электроники, такие как оператор сдвига, приоритетный шифратор или унарный преобразователь обычно остаются в тени. В популярных монографиях по сложности вычислений, таких как [2, 3, 28], даются лишь фрагментарные сведения о

* *Сергеев Игорь Сергеевич* — д.ф.-м.н., нач. лаб., ФГУП «НИИ «Квант», Москва, e-mail: isserg@gmail.com, ORCID: 0000-0002-0622-0843.

Sergeev Igor Sergeevich — Dr.Habil., head of laboratory, FSUE "RDI "Kvant", Moscow.

базовых операторах. Замысел настоящего обзора состоит в том, чтобы представить по возможности полную картину.

Значительную часть излагаемых далее результатов и методов следует отнести к фольклору ввиду простоты и достаточной известности. Поэтому здесь они приводятся без указания литературных источников. В остальных случаях, когда речь идет о результатах, потребовавших определенных усилий, особенно это касается нижних оценок, ссылки на известные автору работы проставлены, как это принято. Некоторые пробелы автору пришлось заполнить самостоятельно, но при этом не прибегая к нетривиальным построениям.

Рассматривается реализация булевых преобразований в модели схем из функциональных элементов, т. е. булевых или логических схем, см., например, [13, 28]. Из стандартных математических моделей она наиболее точно соответствует реальным электронным схемам. Напомним, что *схема над базисом* (множеством функций) $\mathcal{B}$ — это ациклический ориентированный граф, в котором вершины, не имеющие входящих ребер, отмечены как входы, а некоторые вершины отмечены как выходы. Входам приписаны символы переменных или констант базиса $\mathcal{B}$, остальным вершинам (эти вершины называются функциональными элементами) — символы функций из базиса $\mathcal{B}$. Функционирование схемы определяется естественным образом, от входов к выходам: в каждой вершине вычисляется сопоставленная ей функция, аргументами которой служат функции, поступающие по входящим в вершину ребрам. Схема реализует оператор (т. е. систему функций) F, если на выходах схемы вычисляются все компоненты оператора. *Сложность* схемы определяется как число вершин в ее графе без учета входов. Сложность оператора F при реализации схемами над базисом $\mathcal{B}$ определяется как сложность минимальной реализующей его схемы. *Глубина* схемы — это длина (измеряемая в ребрах или функциональных элементах) максимального ориентированного пути, соединяющего вход и выход схемы. По аналогии, глубина оператора F определяется как минимальная глубина реализующей его схемы. Физически сложность примерно соответствует площади схемы, а глубина — задержке распространения сигнала от входов к выходам.

Мы ограничиваем рассмотрение базисом из всех двуместных булевых функций. Сложность схемы Φ над этим базисом обозначим через $\mathsf{C}(\Phi)$, а сложность оператора F — через $\mathsf{C}(F)$. Дополнительно введем функционал $\mathsf{C}_{\log}(F_n)$, обозначающий сложность реализации последовательности операторов F_n схемами глубины $O(\log n)$, где n — число входных переменных. Неформально говоря, это сложность параллельного вычисления оператора. В практической схемотехнике параллельным схемам отдается

предпочтение. Стоит отметить, что рассматриваемые далее операторы достаточно просты и допускают параллельную реализацию.

При проектировании электронных схем обычно доступны более широкие базисы, включающие многовходовые элементы, и даже элементы с несколькими выходами. Однако эффективные методы синтеза, как правило, достаточно универсальны, и могут быть адаптированы к произвольному базису. Напомним, что порядок сложности/глубины оператора в любом полном конечном базисе — один и тот же.

Приводимые оценки сложности характеризуют сложность вычислений в асимптотическом смысле, т.е. при числе входов $n \to \infty$. Однако хорошо известно, что асимптотически эффективные методы синтеза не обязательно дают хороший результат при практически значимых величинах n. Впрочем, упоминаемые далее простые методы обычно неплохо работают уже при самых малых n. Кроме того, при таких значениях n параллельные методы синтеза приводят к схемам с компромиссными характеристиками глубины и сложности.

Далее излагаются сведения о сложности и эффективных способах реализации базовых преобразований. К базовым операциям мы относим префиксные и суффиксные суммы, числовой инкремент/декремент, двусторонний счетчик, счетчик Грея, вычисление переносов, сложение и сравнение двух чисел, максимум/минимум двух чисел, дешифратор, мультиплексор, прямой и циклический сдвиг, шифратор, выделение первой единицы и номера ее позиции, суммирование битов и сравнение суммы с пороговым значением, вычисление ширины единичного блока, перевод в унарную кодировку, обрезка, сортировка битового массива. Оценки сложности перечисленных операций сведены в таблицу 1.

Обзор дополняется примерами применения базовых операций и полезных для синтеза идей на материале параллельных схем 2-селектора, счетчика с сохранением веса, множественного выбора и перестановки пары битов.

При изложении материала используются следующие обозначения:

$\mathbb{B} = \{0, 1\}$;

$\mathbb{B}^n$ — множество булевых строк или векторов длины n; по умолчанию, нумерация битов в строке — с нуля слева направо;

$[\![n]\!]$ — множество целых чисел от 0 до $n-1$, заданных двоичной записью длины $\lceil \log n \rceil$, нумерация разрядов — с нуля справа налево;

$|s|$ — длина булева вектора или строки;

$\nu(X)$ — двоичный вес (число единичных битов или разрядов) булевой строки или числа X;

$\overline{x}$, $x \vee y$, $x \wedge y$ или $x \cdot y$, $x \oplus y$, $x \sim y$ — булевы операции отрицания, дизъюнкции, конъюнкции, сложения по модулю 2, эквивалентности;

$X^\alpha = \bigwedge x_i^{\alpha_i}$ — элементарная конъюнкция вектора переменных $X = [x_1, x_2, \ldots]$, где $\alpha = [\alpha_1, \alpha_2, \ldots] \in \mathbb{B}^{|X|}$; по определению, $x^1 = x$ и $x^0 = \overline{x}$;

« $\|$ » — операция конкатенации строк;

$[\sigma]^m$ — вектор или строка длины m из булевых величин σ.

Все логарифмы далее по тексту имеют основание 2.

Сложность базовых операторов

Префиксные суммы. Оператор $\mathtt{PREF}_n^* : \mathbb{S}^n \to \mathbb{S}^n$ вычисляет семейство префиксных сумм n переменных из полугруппы $(\mathbb{S}, *)$:

$$p_i = x_1 * x_2 * \ldots * x_i, \quad 1 \le i \le n. \tag{1}$$

Дальнейшие применения, за исключением конструкций схем переносов, будут ограничены случаем $\mathbb{S} = \mathbb{B}$ и $* \in \{\vee, \wedge, \oplus\}$. Схемы, вычисляющие систему (1) над базисом $\{*\}$, называются префиксными.

Префиксным схемам посвящена обширная литература, см., например, [15, 24],[28, §3.1]. Мы ограничимся лишь краткими сведениями. Очевидно, оператор $\mathtt{PREF}_n^*$ реализуется схемой из $n-1$ операций $*$, в которой префиксные суммы вычисляются последовательно. Сложность C и глубина D схемы из операций $*$, вычисляющей префиксные суммы n переменных, связаны соотношением $C + D \ge 2n - 2$. Поэтому сложность параллельной схемы не меньше $2n - \Theta(\log n)$. Эта оценка достигается, например, схемами, построенными Ю. П. Офманом [5] (но в литературе их обычно называют схемами Брента—Кунга). В схеме на $n = 2^k$ входах старшая сумма p_n вычисляется полным двоичным деревом T глубины k. Любая недостающая сумма p_i вычисляется как $p_{\bigtriangledown i} * p'$, где $\bigtriangledown i$ обозначает число, получаемое из i обнулением младшего единичного разряда, а p' — подходящая подсумма, вычисленная в дереве T. Легко проверить, что схема имеет глубину $2k - 2$. Схема для 8 входов изображена на рис. 1. Схема с произвольным числом $n < 2^k$ входов получается урезанием схемы на 2^k входах.

В случае булевых операций $*$ все перечисленные выше оценки справедливы для схем над рассматриваемым базисом из двуместных булевых функций, в частности, $\mathsf{C}(\mathtt{PREF}_n^*) = n - 1$ и $\mathsf{C}_{\log}(\mathtt{PREF}_n^*) = 2n - \Theta(\log n)$.

Также встречается задача совместного вычисления префиксных и суффиксных сумм n переменных. Последние определяются как

$$s_i = x_i * x_{i+1} * \ldots * x_n, \quad 1 \le i \le n.$$

Сложность соответствующего оператора $\mathtt{PS}_n^* : \mathbb{S}^n \to \mathbb{S}^{2n-1}$ очевидно равна $2n-3$. Минимальные параллельные префиксно-суффиксные схемы имеют

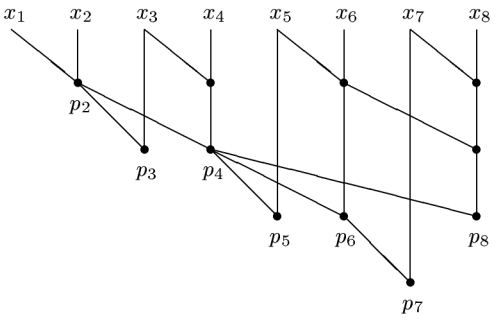

Рис. 1: Префиксная схема Офмана (Брента—Кунга)

сложность $3n - \Theta(\log n)$. Такую схему можно, например, построить совмещением префиксной и суффиксной схем Офмана — дерево, вычисляющее сумму $p_n = s_1$ всех переменных, является общей частью этих двух схем.

Инкрементор. Оператор $\mathtt{INC}_n : [\![2^n]\!] \to [\![2^n]\!]$ увеличивает n-разрядное число на единицу по модулю 2^n.

Верхняя оценка $\mathsf{C}(\mathtt{INC}_n) \le 2n - 2$ при $n \ge 2$ получается элементарно. Если обозначить вход через $X = [x_{n-1}, \ldots, x_0] \in [\![2^n]\!]$, разряды суммы $X + 1 \bmod 2^n = [z_{n-1}, \ldots, z_0]$ определяются формулами[1]

$$z_k = x_k \oplus x_{k-1} \cdot \ldots \cdot x_0. \tag{2}$$

Произведения в этих формулах вычисляются оператором $\mathtt{PREF}^{\wedge}_{n-1}$, после чего выполняется поразрядное сложение векторов длины n по модулю 2. Использование параллельной префиксной схемы приводит к оценке $\mathsf{C}_{\log}(\mathtt{INC}_n) \le 3n - \Theta(\log n)$.

Дополняющую нижнюю оценку $\mathsf{C}(\mathtt{INC}_n) \ge 2n - 2$, вероятно, проще доказать, переходя к рассмотрению оператора $\mathtt{INC}'_n$, вычисляющего сумму $X + 1$ целиком. Легко проверить, что $\mathsf{C}(\mathtt{INC}'_n) = \mathsf{C}(\mathtt{INC}_n) + 1$ при $n \ge 2$. Далее, $\mathsf{C}(\mathtt{INC}'_n) = 2n - 1$, поскольку при подстановке $x_{n-1} = 0$ в схему для $\mathtt{INC}'_n$ сложность уменьшается минимум на 2, а полученная схема вычисляет оператор $\mathtt{INC}'_{n-1}$.

[1] Формула справедлива и при $k = 0$, если принять соглашение, что конъюнкция с пустым множеством операндов равна 1.

Схемы декремента, вычисляющие $X-1 \bmod 2^n = [z_{n-1},\ldots,z_0]$, устроены двойственным образом: разряды разности определяются формулами

$$z_k = x_k \oplus \overline{x_{k-1}} \cdot \ldots \cdot \overline{x_0}. \tag{3}$$

Двусторонний счетчик. Оператор $\mathtt{UDC}_n : [\![2^n]\!] \times \mathbb{B} \to [\![2^n]\!]$ в зависимости от управляющего входа $\sigma \in \mathbb{B}$ выполняет либо инкремент (при $\sigma = 1$), либо декремент (при $\sigma = 0$) n-разрядного числа по модулю 2^n.

Оценки $\mathsf{C}(\mathtt{UDC}_n) \le 3n-3$ при $n \ge 2$ и $\mathsf{C}_{\log}(\mathtt{UDC}_n) \le 4n - \Theta(\log n)$ получаются совмещением схем инкрементора и декрементора из предыдущего пункта. Разряды результата вычисляются по формулам, объединяющим (2) и (3):

$$z_k = x_k \oplus x_{k-1}^{\sigma} \cdot \ldots \cdot x_0^{\sigma}.$$

Булевы степени x_k^{σ} вычисляются просто как $\sigma \sim x_k$, $k = 0,\ldots,n-2$. Далее используется оператор $\mathtt{PREF}^{\wedge}_{n-1}$, затем выполняется поразрядное сложение векторов длины n.

Счетчик Грея. Оператор $\mathtt{GRC}_n : \mathbb{B}^n \to \mathbb{B}^n$ по булевой строке длины n вычисляет следующую за ней циклически в стандартной кодировке Грея. Иначе говоря, это инкрементор в коде Грея. Разряды строк в этом пункте нумеруются справа налево.

Напомним, что последовательность n-разрядных строк Грея G_n можно задать рекурсивно: $G_1 = (0,\,1)$ и далее $G_{k+1} = (0 \,\|\, G_k,\, 1 \,\|\, G_k^R)$, где G_k^R — последовательность G_k, записанная в обратном порядке; операция конкатенации применяется построчно. Основное свойство последовательностей Грея: следующая строка отличается от предыдущей ровно в одной позиции. Подробнее см., например, в [10, Гл. 13].

Верхние оценки $\mathsf{C}(\mathtt{GRC}_n) \le 4n-7$ и $\mathsf{C}_{\log}(\mathtt{GRC}_n) \le 6n - \Theta(\log n)$ получаются переводом в двоичный номер строки и обратно. Пусть $\mathrm{bin}_n(X)$ означает двоичную запись номера строки X в последовательности G_n. Тогда $\mathtt{GRC}_n(X) = \mathrm{bin}_n^{-1}(\mathtt{INC}_n(\mathrm{bin}_n(X)))$. Теперь, обозначая $[y_{n-1},\ldots,y_0] = \mathrm{bin}_n(x_{n-1},\ldots,x_0)$, легко установить, что $y_{n-1} = x_{n-1}$ и при любом $i \le n-2$

$$y_i = x_i \oplus x_{i+1} \oplus \ldots \oplus x_{n-1}, \qquad x_i = y_{i+1} \oplus y_i.$$

Поэтому преобразование bin_n выполняется оператором $\mathtt{PREF}^{\oplus}_n$, а bin_n^{-1} сводится к поразрядному сложению строк длины $n-1$. Чуть удобнее вместо y_i вычислять дополняющие биты $\overline{y_i}$. При последовательной реализации оператора $\mathtt{INC}_n(X)$ два младших разряда вычислять не нужно — они равны x_0 и $\overline{y_0}$ — остальная часть схемы инкремента имеет сложность $2n-4$. Еще одна операция экономится на последнем шаге преобразования в код Грея: младший разряд результата просто равен $\overline{y_1}$.

Оператор переносов. Оператор $\mathtt{CAR}_n : (\mathbb{B}^n)^2 \to \mathbb{B}^n$ вычисляет систему функций переноса

$$c_1 = x_0, \qquad c_{i+1} = x_i \oplus y_i c_i, \quad i = 1, \ldots, n-1, \tag{4}$$

от булевых переменных $x_0, x_1, \ldots, x_{n-1}$ и $y_0, y_1, \ldots, y_{n-1}$.

Прямо из определения (4) очевидно следует $\mathsf{C}(\mathtt{CAR}_n) = 2n - 2$. Оценка $\mathsf{C}_{\log}(\mathtt{CAR}_n) \leq 5n - \Theta(\log n)$ доказывается сведением к вычислению системы префиксных сумм. Пусть бинарная операция $\star$ на множестве булевых векторов длины 2 определяется как $[a_1, b_1] \star [a_2, b_2] = [a_2 \oplus b_2 a_1, b_2 b_1]$. Несложно убедиться, что введенная операция ассоциативна, значит, $(\mathbb{B}^2, \star)$ — полугруппа. С обозначением $p_i = y_{i-1} \cdot \ldots \cdot y_1$ система (4) трансформируется в систему префиксных сумм

$$[c_1, p_1] = [x_0, 1], \qquad [c_{i+1}, p_{i+1}] = [c_i, p_i] \star [x_i, y_i], \quad i = 1, \ldots, n-1.$$

Заметим, что собственно произведения p_i при реализации переносов вычислять не нужно. Тогда для сложности параллельных схем переносов имеем оценку $\mathsf{C}_{\log}(\mathtt{CAR}_n) \leq \mathsf{C}_{\log}(\mathtt{PREF}^{\star}_n) - (n-1)$. Остается заметить, что сложность операции $\star$ равна 3.

Идея вычисления переносов параллельными префиксными схемами возникла не позднее 1960-х гг., например, в [5, 25].

Сумматор. Оператор $\mathtt{ADD}_n : [\![2^n]\!]^2 \to [\![2^{n+1}]\!]$ вычисляет сумму двух n-разрядных чисел A и B.

Верхние оценки $\mathsf{C}(\mathtt{ADD}_n) \leq 5n - 3$ и $\mathsf{C}_{\log}(\mathtt{ADD}_n) \leq 8n - \Theta(\log n)$ получаются добавлением к схемам переносов из предыдущего пункта двух слоев из $3n - 1$ элементов[2]. Введем обозначения $A = [a_{n-1}, \ldots, a_0]$, $B = [b_{n-1}, \ldots, b_0]$ и $A + B = [z_n, \ldots, z_0]$. Положим $x_i = a_i \wedge b_i$, $y_i = a_i \oplus b_i$. Пусть c_i определяются согласно (4). Тогда $z_0 = y_0$, $z_n = c_n$ и $z_i = y_i \oplus c_i$ при прочих i.

Н. П. Редькин [6] показал, что первая из двух оценок точна: в действительности, $\mathsf{C}(\mathtt{ADD}_n) = 5n - 3$.

Мы не рассматриваем отдельно операцию вычитания чисел, поскольку отрицательные числа обычно задаются дополнительным кодом, в котором вычитание и сложение выполняются единообразно при помощи сумматоров, см., например, [2, разд. 4.1], [28, §3.1].

Компаратор. Функция $\mathtt{CMP}_n : [\![2^n]\!]^2 \to \mathbb{B}$ выполняет сравнение двух n-разрядных чисел A, B, т. е. вычисляет значение предиката $A > B$.

Верхняя оценка $\mathsf{C}(\mathtt{CMP}_n) \leq 4n - 3$ получается непосредственно. Обозначим $A = [a_{n-1}, \ldots, a_0]$, $B = [b_{n-1}, \ldots, b_0]$, а также $x_i = a_i \wedge \overline{b_i}$,

[2] В работах [19, 28] оценки сложности параллельного сумматора приводятся в форме $\mathsf{C}_{\log}(\mathtt{ADD}_n) \leq 8n + O(1)$.

$y_i = a_i \sim b_i$. Тогда $\mathtt{CMP}_n(A,B) = c_n$, где c_n определяется согласно (4). Поэтому $\mathsf{C}(\mathtt{CMP}_n) \le 2n-1+\mathsf{C}(\mathtt{CAR}_n)$, учитывая, что y_0 вычислять не надо.

Аналогично устанавливается оценка $\mathsf{C}_{\log}(\mathtt{CMP}_n) \le 2n-1+\mathsf{C}_{\log}(c_n) \le 5n-\Theta(\log n)$. Вычислим c_n при помощи параллельного префиксного дерева глубины d за $n-1$ операций $\star$, при этом d элементов, вычисляющих вторые компоненты префиксных сумм (произведения входных переменных), можно сократить.

Нестрогое сравнение реализуется подобным образом, поскольку предикат $A \ge B$ является отрицанием предиката $B > A$.

Расширенный оператор $\mathtt{CMP}^*_n : [\![2^n]\!]^2 \to \mathbb{B}^2$ дополнительно вычисляет признак $A = B$, т. е. произведение $y_0 \cdot y_1 \cdot \ldots \cdot y_{n-1}$. Его сложность оценивается как $\mathsf{C}_{\log}(\mathtt{CMP}^*_n) \le 5n-3$: в описанную выше параллельную схему компаратора надо добавить вычисление y_0 и всех вторых компонент префиксных сумм — из этих компонент в итоге и получается искомое произведение.

Максимум двух чисел. Оператор $\mathtt{MAX}_n : [\![2^n]\!]^2 \to [\![2^n]\!]$ вычисляет максимум из двух n-разрядных чисел.

Верхние оценки $\mathsf{C}(\mathtt{MAX}_n) \le 6n-3$ и $\mathsf{C}_{\log}(\mathtt{MAX}_n) \le 7n-\Theta(\log n)$ получаются присоединением к схемам компаратора из предыдущего пункта слоя из $2n$ булевых операций. Если обозначить $c = \mathtt{CMP}_n(A,B)$, то $\max(A,B) = \overline{(A \sim B)} \cdot [c]^n \oplus B$, где операции в последней формуле — поразрядные. Напомним, что вектор $A \sim B$, за исключением его младшего разряда, уже вычислен схемой компаратора. Поэтому для определения каждого разряда результата достаточно дополнительно выполнить по две операции, при этом старший разряд максимума просто равен $a_{n-1} \vee b_{n-1}$.

Для того, чтобы вместе с максимумом вычислить минимум, достаточно добавить в схему еще n элементов, поскольку $\min(A,B) = \overline{(A \sim B)} \cdot [c]^n \oplus A$.

Дешифратор. Оператор $\mathtt{DEC}_n : [\![n]\!] \to \mathbb{B}^n$ реализует булеву строку длины n с единственной единицей на заданной позиции. Компоненты оператора — элементарные конъюнкции X^α набора переменных X, где вектор α пробегает множество двоичных представлений чисел от 0 до $n-1$.

Верхняя оценка $\mathsf{C}_{\log}(\mathtt{DEC}_n) \le n+\Theta(\sqrt{n})$ получается тривиально делением множества переменных пополам: если $X = [X_2, X_1]$, то $X^{\alpha_2 \,\|\, \alpha_1} = X_2^{\alpha_2} \cdot X_1^{\alpha_1}$. Тогда $\mathsf{C}_{\log}(\mathtt{DEC}_n) \le n + \mathsf{C}_{\log}(\mathtt{DEC}_{2^k}) + \mathsf{C}_{\log}(\mathtt{DEC}_{\lceil n2^{-k} \rceil})$, где $k = |X_1|$. Остается выбрать $k \approx \log n/2$.

Нижнюю оценку в форме $\mathsf{C}(\mathtt{DEC}_n) \ge n+\Theta(\sqrt{n})$ легко установить, заметив, что множество элементов схемы, непосредственно предшествующих выходам, имеет мощность не менее $\sqrt{n}$.

Расширенный оператор $\mathtt{DEC}^*_n : [\![n]\!] \times \mathbb{B} \to \mathbb{B}^n$ имеет дополнительный информационный вход $y \in \mathbb{B}$ и вычисляет вектор с битом y в заданной позиции и остальными нулями. Такой оператор часто называется демультиплексором. Его сложность отличается от сложности дешифратора не более чем на 2: достаточно в схеме дешифратора вместо младшей переменной x и ее отрицания использовать соответственно xy и $\overline{x} \cdot y$.

Мультиплексоры. Функция $\mathtt{MUX}_n : [\![n]\!] \times \mathbb{B}^n \to \mathbb{B}$ реализует выбор одной из n информационных переменных по ее номеру (иначе называемому адресом). Эту функцию называют $(n, 1)$-мультиплексором, а также селектором, функцией выбора или функцией обращения к памяти.

Лучшая известная верхняя оценка $\mathsf{C}_{\log}(\mathtt{MUX}_n) \leq 2n + O(\sqrt{n})$ получена П. Клейном и М. Патерсоном в [18]. Адресный вход представим как $X = [X_2, X_1]$, где $|X_1| = q$. Строку информационных переменных разобьем на блоки длины 2^q: $Y = Y_0 \,\|\, Y_1 \,\|\, \ldots \,\|\, Y_{p-1}$, $p = \lceil n/2^q \rceil$ (последний блок может быть короче). Разложим функцию $\mathtt{MUX}_n$ по переменным X_2:

$$\mathtt{MUX}_n(X; Y) = \bigvee_{\alpha=0}^{p-1} X_2^\alpha \cdot \mathtt{MUX}_{|Y_\alpha|}(X_1; Y_\alpha). \tag{5}$$

Внутренние мультиплексорные функции вычисляются по формулам

$$\mathtt{MUX}_{|Y_\alpha|}(X_1; Y_\alpha) = \bigvee_{\beta=0}^{|Y_\alpha|-1} y_{\alpha,\beta} \cdot X_1^\beta, \tag{6}$$

где подразумевается введение нумерации с двумя индексами на множестве переменных $Y = \{y_{\alpha,\beta}\}$.

Все элементарные конъюнкции групп переменных X_1 и X_2 вычисляются со сложностью порядка $2^q + p$ при помощи дешифраторов $\mathtt{DEC}_{2^q}$ и $\mathtt{DEC}_p$. Еще $2n + p$ операций достаточно для завершения вычислений по формулам (5), (6). Остается выбрать $q \approx \log n/2$.

Указанная оценка асимптотически точна: В. Пауль [23] установил, что $\mathsf{C}(\mathtt{MUX}_n) \geq 2n - 2$.

Очень часто возникает более общая задача — реализация (n, k)-мультиплексора $\mathtt{MUX}^k_n : [\![n]\!] \times (\mathbb{B}^k)^n \to \mathbb{B}^k$, информационными входами которого являются k-разрядные числа или булевы векторы. Схема для оператора $\mathtt{MUX}^k_n$ получается параллельным объединением описанных выше схем $(n, 1)$-мультиплексоров для каждого из k разрядов. Эти схемы имеют общую часть $\mathtt{DEC}_{2^q}(X_1)$ и $\mathtt{DEC}_p(X_2)$. При $q = \lceil \min(\log n, \log(kn)/2) \rceil$ получается оценка $\mathsf{C}_{\log}(\mathtt{MUX}^k_n) \leq 2kn + O(\sqrt{kn})$.

Операторы сдвига. Оператор $\mathtt{CYC}_{k,n} : [\![k]\!] \times \mathbb{B}^n \to \mathbb{B}^n$ выполняет циклический сдвиг булева вектора длины n на $X \in [\![k]\!]$ позиций[3].

[3]Направление сдвига для оценок сложности несущественно.

Верхняя оценка $\mathsf{C}_{\log}(\mathtt{CYC}_{k,n}) \le 3\lceil\log k\rceil n$ тривиальна. Соответствующая схема имеет вид l-кратной композиции $\mathtt{MUX}_2^n \circ \ldots \circ \mathtt{MUX}_2^n$, где $l = \lceil\log k\rceil$. Запишем $X = [x_{l-1}, \ldots, x_1, x_0]$. Первая подсхема выполняет циклический сдвиг на x_0, вторая — на $2x_1$, третья — на $4x_2$, и т. д.

Аналогично реализуется оператор обычного сдвига $\mathtt{SFT}_{k,n} : [\![k]\!] \times \mathbb{B}^n \to \mathbb{B}^{n+k-1}$. Справедлива оценка $\mathsf{C}_{\log}(\mathtt{SFT}_{k,n}) \le 3\lceil\log k\rceil n - 2(k-1)$, учитывающая, что $k-1$ штук (2, 1)-мультиплексоров в описанной выше схеме упрощаются до одной операции.

Шифратор. Оператор $\mathtt{ENC}_n : \mathbb{B}^n \to [\![n]\!]$ — это линейный булев оператор с матрицей U_n размера $\lceil\log n\rceil \times n$, в столбцах которой последовательно записаны числа от 0 до $n-1$ в двоичном представлении[4]. У входного вектора веса 1 шифратор определяет номер единичной позиции. Поэтому на множестве векторов веса 1 шифратор $\mathtt{ENC}_n$ является обратным преобразованием к $\mathtt{DEC}_n$.

Известно, что $\mathsf{C}(\mathtt{ENC}_n) = \mathsf{C}_{\log}(\mathtt{ENC}_n) = 2(n - \lceil\log n\rceil - 1)$. Нижняя оценка доказана А. В. Чашкиным [12]. Верхнюю оценку можно получить элементарно, используя известное соотношение между сложностью линейных операторов с транспонированными матрицами (принцип транспонирования [4]): сложность оператора с матрицей U_n^T тривиально равна $n - \lceil\log n\rceil - 1$. Однако нетрудно и явно описать конструкцию соответствующих схем. Семейство схем Φ_n, вычисляющих $\mathtt{ENC}_n$, построим рекурсивно вместе со схемами Φ'_n, которые реализуют преобразования $\mathtt{ENC}'_n$ с матрицами U_n, дополненными строками из всех единиц. Пусть $n = 2^k + p \le 2^{k+1}$. Разобьем вектор X длины n на две части X_1, X_2 длины 2^k и p. Результат $\Phi_n(X)$ получается из $\Phi_{2^k}(X_1)$ и $\Phi'_p(X_2)$ при помощи дополнительных $\lceil\log p\rceil$ булевых операций сложения, а $\Phi'_n(X)$ получается из $\Phi'_{2^k}(X_1)$ и $\Phi'_p(X_2)$ при помощи $\lceil\log p\rceil + 1$ операций. Рекуррентные соотношения

$$\mathsf{C}(\Phi_n) = \mathsf{C}(\Phi_{2^k}) + \mathsf{C}(\Phi'_p) + \lceil\log p\rceil, \quad \mathsf{C}(\Phi'_n) = \mathsf{C}(\Phi'_{2^k}) + \mathsf{C}(\Phi'_p) + \lceil\log p\rceil + 1$$

разрешаются как $\mathsf{C}(\Phi_n) = 2(n - \lceil\log n\rceil - 1)$ и $\mathsf{C}(\Phi'_n) = 2n - \lceil\log n\rceil - 2$ с учетом начальных условий $\mathsf{C}(\Phi_1) = \mathsf{C}(\Phi'_1) = 0$. Глубина схем Φ_n и Φ'_n равна $\lceil\log n\rceil - 1$ и $\lceil\log n\rceil$ соответственно. Подробнее см. в [13].

В альтернативном варианте определения, шифратор $\mathtt{ENC}^*_n$ — это частичный булев оператор, определенный только на множестве векторов веса 1 и совпадающий с $\mathtt{ENC}_n$ на этом множестве. Следуя схеме доказательства нижней оценки из [13], несложно проверить, что ослабление определения не дает существенного выигрыша: $\mathsf{C}(\mathtt{ENC}^*_n) = 2n - \Theta(\log n)$.

[4]При $n = 2^k$ указанная матрица является с точностью до наличия нулевого столбца проверочной матрицей кода Хэмминга.

Унарная кодировка. Оператор $\mathtt{UN}_n : [\![n+1]\!] \to \mathbb{B}^n$ переводит число из стандартной двоичной записи в унарную — в ней число k записывается строкой, начинающейся с k единиц и продолжаемой нулями.

Верхнюю оценку $\mathsf{C}(\mathtt{UN}_n) \le 2n + O(\sqrt{n})$ просто получить при помощи схемы типа $\mathtt{PREF}^{\vee} \circ \mathtt{DEC}$. Оператор $\mathtt{DEC}_{n+1}(x)$ вычисляет вектор с одной единицей в позиции x. Отбрасывая младшую компоненту вектора, вычисляем суффиксные суммы оставшихся компонент.

Указанная оценка усиливается до $\mathsf{C}_{\log}(\mathtt{UN}_n) \le 2n + O(\sqrt{n})$. Сначала индуктивно построим схемы для $\mathtt{UN}_{2^k-1}$ сложности $2(2^k - k - 1)$ и глубины $k-1$. Пусть булева строка $S = [s_1, \ldots, s_{2^k-1}]$ получается как $S = \mathtt{UN}_{2^k-1}(X)$, где X — вектор булевых переменных длины k. Тогда

$$\mathtt{UN}_{2^{k+1}-1}(y, X) = \begin{cases} S \parallel [0]^{2^k}, & y = 0 \\ [1]^{2^k} \parallel S, & y = 1 \end{cases} = \\ = [s_1 \vee y,\ \ldots,\ s_{2^k-1} \vee y,\ y,\ s_1 \cdot y,\ \ldots,\ s_{2^k-1} \cdot y].$$

Таким образом, схема для $\mathtt{UN}_{2^{k+1}-1}$ строится из схемы для $\mathtt{UN}_{2^k-1}$ добавлением слоя из $2(2^k-1)$ элементов конъюнкции и дизъюнкции, откуда с учетом $\mathsf{C}(\mathtt{UN}_1) = 0$ получаются требуемые оценки.

Схему для $\mathtt{UN}_n$ построим блочным методом. Пусть $X = [X_2, X_1]$, где $|X_1| = k \approx \log n/2$. Обозначим $p = \lceil (n+1)/2^k \rceil$. Результат $\mathtt{UN}_n(X)$ можно записать в блочном виде как $B_0 \parallel B_1 \parallel \ldots \parallel B_{p-1}$, где все строки B_i, за исключением последней неполной[5], имеют длину 2^k. Вычислим $\mathtt{DEC}_p(X_2) = [a_0, a_1, \ldots, a_{p-1}]$ — это вектор с индикатором позиции первого не сплошь единичного блока. Такой блок имеет вид $S = \mathtt{UN}_{2^k-1}(X_1) \parallel 0$ (укороченный в случае крайнего блока), слева от него блоки сплошь из единиц, справа — сплошь из нулей. При помощи оператора $\mathtt{PREF}_p^{\vee}$ вычислим префиксные суммы $b_i = a_0 \vee a_1 \vee \ldots \vee a_i$. Тогда

$$B_i = \begin{cases} [1]^{2^k}, & a_i = b_i = 0 \\ S, & a_i = b_i = 1 \\ [0]^{2^k}, & a_i = 0, b_i = 1 \end{cases} = S \cdot [a_i]^{2^k} \vee [\overline{b_i}]^{2^k}, \qquad (7)$$

где в правой части операции с булевыми строками выполняются поразрядно; при $i = p-1$ блок урезан. Окончательно, схема состоит из подсхем $\mathtt{DEC}_p$, $\mathtt{PREF}_p$, $\mathtt{UN}_{2^k-1}$ сложности $O(\sqrt{n})$ и слоя из $2n$ операций, предусмотренных в (7).

[5] Ее длина — от 0 до $2^k - 1$.

Обратное преобразование $\mathtt{UN}_n^{-1} : \mathbb{B}^n \to [\![n+1]\!]$ из унарного кода в двоичный выполняется несложно: $\mathsf{C}_{\log}(\mathtt{UN}_n^{-1}) = n - 1$. Разряды числа $[y_k, \ldots, y_0] = \mathtt{UN}_n^{-1}(s_1, \ldots, s_n)$ определяются формулами

$$y_j = \bigoplus_{i \geq 1} s_{i \cdot 2^j}.$$

Все необходимые суммы можно вычислить деревом глубины $\lceil \log n \rceil$ из $n - 1$ элементов $\oplus$.

Оператор обрезки. Оператор $\mathtt{TRN}_n : [\![n+1]\!] \times \mathbb{B}^n \to \mathbb{B}^n$ в булевой строке длины n сохраняет первые k бит, а остаток строки заполняет нулями.

Верхняя оценка $\mathsf{C}_{\log}(\mathtt{TRN}_n) \leq 3n + O(\sqrt{n})$ получается автоматически ввиду $\mathtt{TRN}_n(k; X) = \mathtt{UN}_n(k) \wedge X$ (конъюнкция выполняется поразрядно).

Индикатор первой единицы. Оператор $\mathtt{FOI}_n : \mathbb{B}^n \to \mathbb{B}^n \times \mathbb{B}$ оставляет в булевой строке длины n самую первую единицу, заменяя остальные нулями, и дополнительно вычисляет признак наличия единицы в строке.

Легко видеть, что оператор преобразует строку $X = [x_0, x_1, \ldots, x_{n-1}]$ в $Y = [y_0, y_1, \ldots, y_{n-1}; z]$, где $y_k = \overline{(x_0 \vee \ldots \vee x_{k-1})} \cdot x_k$ и $z = x_0 \vee \ldots \vee x_{n-1}$. Для его вычисления достаточно к схеме, вычисляющей $\mathtt{PREF}_n^{\vee}(X)$, добавить слой из $n - 1$ булевых операций типа $\overline{a} \wedge b$. Как следствие, $\mathsf{C}(\mathtt{FOI}_n) \leq 2n - 2$ и $\mathsf{C}_{\log}(\mathtt{FOI}_n) \leq 3n - \Theta(\log n)$.

Первая оценка точна: несложно проверить, что схема, реализующая $\mathtt{FOI}_n$, после подстановки $x_{n-1} = 0$ вычисляет $\mathtt{FOI}_{n-1}$ и при этом в ней сокращаются минимум два элемента.

Приоритетный шифратор. Оператор $\mathtt{PENC}_n : \mathbb{B}^n \to [\![n]\!] \times \mathbb{B}$ определяет номер позиции первой единицы в булевой строке длины n и дополнительно вычисляет признак наличия единицы в строке[6].

Верхние оценки $\mathsf{C}(\mathtt{PENC}_n) \leq 2n - 2$ и $\mathsf{C}_{\log}(\mathtt{PENC}_n) \leq 3n - \Theta(\log n)$ достигаются схемами вида $\mathtt{UN}^{-1} \circ \mathtt{PREF}$. Действительно, номер первой единичной позиции строки X можно найти как $\mathtt{UN}_n^{-1}(\overline{\mathtt{PREF}_n^{\vee}(X)})$, где операция отрицания применяется поразрядно. Попутно внутренний оператор $\mathtt{PREF}_n^{\vee}$ вычисляет признак наличия единицы. При $n \geq 3$ справедливо даже $\mathsf{C}(\mathtt{PENC}_n) \leq 2n - 3$: номер позиции корректно вычисляется как $\mathtt{UN}_{n-1}^{-1}(\overline{\mathtt{PREF}_{n-1}^{\vee}(X')})$, где X' — строка X без последнего символа.

Нижнюю оценку можно указать в виде $\mathsf{C}(\mathtt{PENC}_n) \geq \mathsf{C}(\mathtt{ENC}_n^*) = 2n - \Theta(\log n)$, поскольку оператор $\mathtt{PENC}_n$ является доопределением оператора $\mathtt{ENC}_n^*$.

Суммирование битов. Оператор $\mathtt{SUM}_n : \mathbb{B}^n \to [\![n+1]\!]$ вычисляет арифметическую сумму n булевых переменных.

[6] Это частичный булев оператор: номер позиции не определен при нулевом входе.

Сложность суммирования битов достаточно хорошо изучена. Эффективные схемы суммирования битов строятся из подсхем-компрессоров. Компрессор преобразует несколько битовых входов в меньшее число выходов с сохранением суммы (с учетом разряда). Пример схемы суммирования, составленной из (3, 2)-компрессоров $\Sigma_{3,2}$, вычисляющих сумму трех бит, показан на рис. 2.

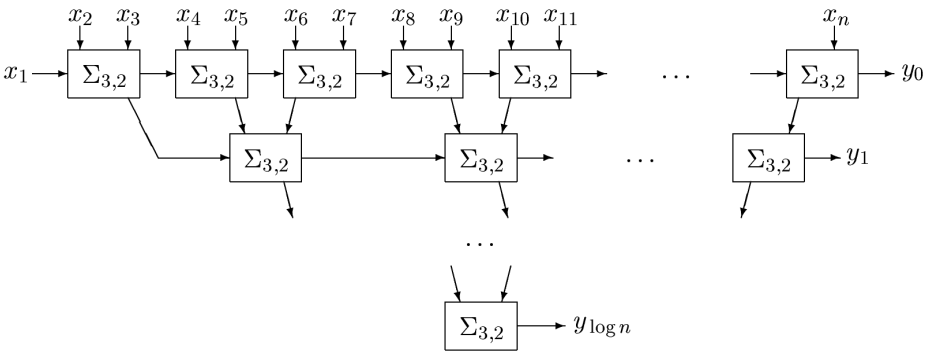

Рис. 2: Стандартная схема суммирования n бит

В работе [16] из специальных (5, 3)-компрессоров сконструирована схема сложности $4.5n - \Theta(\log n)$. Ценой увеличения сложности до $4.5n + o(n)$ схему можно сделать параллельной [7]. Для этого, например, можно разбить входы на группы по $\log n$ штук, вычислить суммы в группах методом [16], затем сложить групповые суммы при помощи любой параллельной схемы компрессоров. Первый этап имеет сложность $4.5n - o(n)$, а второй — $o(n)$. Таким образом, $\mathsf{C}(\mathtt{SUM}_n) \le 4.5n - \Theta(\log n)$ и $\mathsf{C}_{\log}(\mathtt{SUM}_n) \le 4.5n + o(n)$. О построении параллельных схем из компрессоров подробно рассказывается в [22], см. также [28, §3.2].

Нижнюю оценку можно указать в виде $\mathsf{C}(\mathtt{SUM}_n) \ge 2.5n + \Theta(\log n)$. Она получается сопоставлением доказанной Л. Стокмейером [26] оценки $2.5n - O(1)$, справедливой для каждой компоненты оператора $\mathtt{SUM}_n$, кроме разве что младшей и нескольких старших, и простого соотношения [20]

для сложности системы функций:

$$\mathsf{C}(f_1, \ldots, f_k) \geq \min_i \mathsf{C}(f_i) + k - 1. \tag{8}$$

Подсчет числа единиц в блоке. Оператор $\mathtt{BW}_n : \mathbb{B}^n \to [\![n+1]\!]$ для булевой строки длины n, которая содержит только один блок из единиц, вычисляет длину этого блока.

Верхние оценки $\mathsf{C}(\mathtt{BW}_n) \leq 4n - \Theta(\log n)$ и $\mathsf{C}_{\log}(\mathtt{BW}_n) \leq 4n + o(n)$ доказываются методом компрессоров. Базовая схема строится как на рис. 2 из подсхем-компрессоров $\Sigma_{3,2} : (a, b, c) \to [u, v]$, суммирующих тройки бит: $a + b + c = 2u + v$. При условии, что единицы на входе расположены одним сплошным блоком, такие подсхемы можно реализовать со сложностью 4: $u = b(a \vee c)$, $v = a \oplus b \oplus c$. Легко видеть, что если на вход схемы рис. 2 подается строка с одним единичным блоком, то единицы на входе каждого внутреннего компрессора будут смежны, иначе говоря, входной вектор (1, 0, 1) не встретится. Общая сложность описанной схемы равна $4n - \Theta(\log n)$. Параллельная схема строится как указано в предыдущем пункте.

Пороговые симметрические функции. Оператор $\mathtt{THR}_n^k : \mathbb{B}^n \to \mathbb{B}$ сравнивает число единиц в булевой строке X длины n с порогом k и вычисляет значение предиката $\nu(X) \geq k$.

Верхние оценки $\mathsf{C}(\mathtt{THR}_n^k) \leq 4.5n + O(\log n)$ и $\mathsf{C}_{\log}(\mathtt{THR}_n^k) \leq 4.5n + o(n)$ вытекают непосредственно из соответствующих оценок сложности оператора суммирования битов; к схеме для $\mathtt{SUM}_n$ достаточно присоединить схему сравнения суммы с порогом. Л. Стокмейер [26] доказал нижнюю оценку $\mathsf{C}(\mathtt{THR}_n^k) \geq 2n + \min\{k - 2, n - k - 1\} - 3$.

Аналогично строятся схемы для других симметрических функций, например, периодических.

Для постоянного порога k специальные методы синтеза позволяют получить лучшие оценки сложности. В частности, $\mathsf{C}(\mathtt{THR}_n^2) = \mathsf{C}_{\log}(\mathtt{THR}_n^2) = 2n + \Theta(\sqrt{n})$, $\mathsf{C}_{\log}(\mathtt{THR}_n^3) \leq 3n + O(\log n)$, а в общем виде для $2^{p-1} < k \leq 2^p$ имеет место $\mathsf{C}_{\log}(\mathtt{THR}_n^k) \leq (4.5 - 2^{2-p})n + \theta_k(n)$, где $\theta_k(n) = O(\sqrt{n})$ или $\theta_k(n) = O(\log n)$, см. [8].

Сортировка. Оператор $\mathtt{SORT}_n : \mathbb{B}^n \to \mathbb{B}^n$ упорядочивает строку битов в порядке возрастания (нули слева, единицы справа).

Верхние оценки $\mathsf{C}(\mathtt{SORT}_n) \leq 6.5n + O(\sqrt{n})$ и $\mathsf{C}_{\log}(\mathtt{SORT}_n) \leq 6.5n + o(n)$ достигаются на схемах типа $\mathtt{UN} \circ \mathtt{SUM}$. Сначала при помощи оператора $\mathtt{SUM}_n$ вычисляется число единиц в строке, затем посредством $\mathtt{UN}_n$ — унарное представление этого числа.

Нижняя оценка $\mathsf{C}_{\log}(\mathtt{SORT}_n) \geq 3n - 6$ получается сопоставлением доказанной Б. М. Клоссом [1] нижней оценки $2n - 3$, справедливой для

любой компоненты оператора, кроме младшей и старшей (минимума и максимума), и оценки (8).

Некоторые приложения

В этом разделе мы рассмотрим несколько примеров применения рассмотренных выше конструкций к построению параллельных схем более узкого назначения. Первый пример иллюстрирует применение параллельных префиксно-суффиксных схем. Второй пример опирается на эффективную реализацию унарно-бинарных преобразований. Третий пример иллюстрирует принцип массового производства и приводит к четвертому примеру.

2-селектор. Оператор $\mathtt{TOI}_n : \mathbb{B}^n \to \mathbb{B}^n \times \mathbb{B}$ выделяет в булевой строке длины n две единицы, заменяя остальные нулями, и дополнительно вычисляет признак наличия единицы в строке. Этот оператор является расширением оператора $\mathtt{FOI}_n$ и применяется для выделения в наборе из n каналов данных двух активных — отмеченных единицами, см., например, [14].

Оценка $\mathsf{C}_{\log}(\mathtt{TOI}_n) \le 5n - \Theta(\log n)$ достигается на схеме, выделяющей две крайних единицы с разных сторон. При помощи оператора $\mathtt{PS}^{\vee}_n$ по исходной строке $[x_1, \ldots, x_n]$ вычисляются строка префиксных сумм $[p_1, \ldots, p_n] = [0]^k[1]^{n-k}$ и строка суффиксных сумм $[s_1, \ldots, s_n] = [1]^l[0]^{n-l}$, где $k+1$ и l — позиции первой и последней единицы (если единицы есть). При этом $p_n = s_1$ служит признаком наличия единиц. Результирующая строка $[z_1, \ldots, z_n]$ вычисляется при помощи поразрядных операций $z_i = x_i \cdot (\overline{p_{i-1}} \vee \overline{s_{i+1}})$.

Счетчик с сохранением веса. Оператор $\mathtt{NCK}_n : [\![2^n]\!] \to [\![2^n]\!]$ вычисляет следующее за данным в порядке возрастания число с таким же двоичным весом. Если большего числа нет, то выход совпадает со входом. Этот оператор, в частности, рассматривался в [21] в связи с приложениями в области обработки изображений — там он назван $\binom{n}{k}$-счетчиком. Сложность параллельной реализации оператора $\mathtt{NCK}_n$ в [21] указана как $O(n)$, из описания схемы извлекается оценка около $16n$.

Верхнюю оценку сложности можно уточнить до $\mathsf{C}_{\log}(\mathtt{NCK}_n) \le 13n + o(n)$. Приписав к произвольному отличному от нуля n-разрядному числу дополнительный ноль слева, это число можно однозначно представить в виде $S \,\|\, 0\,[1]^j[0]^i$, где $i \ge 0$, $j \ge 1$ и S — непустая подстрока в случае, когда число не максимально среди чисел одинакового веса. Тогда оператор $\mathtt{NCK}_n$ преобразует немаксимальное число как $S \,\|\, 0\,[1]^j[0]^i \to S \,\|\, 1\,[0]^{i+1}[1]^{j-1}$. Последовательность вычислений, приводящих к требуемому результату, изображена на рис. 3. Через $*$ обозначены несущественные части строк, `bit[]` означает соответствующую поразрядную операцию. Аргументы и

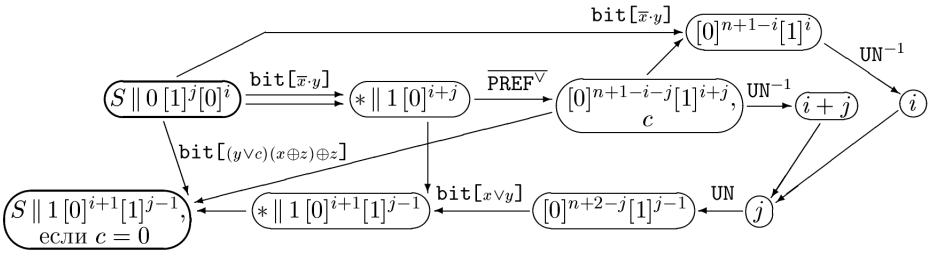

Рис. 3: Диаграмма алгоритма вычислений в схеме счетчика с сохранением веса

результаты некоторых операций сдвинуты на 1. Дополнительные биты показаны для общности записи, в вычислениях они не используются. Схема состоит из подсхем, реализующих операторы $\mathtt{PREF}^\vee_n$, $\mathtt{UN}_n$, $\mathtt{UN}^{-1}_n$ (дважды), семь поразрядных операций с векторами длины n и вычитание $\log n$-разрядных чисел. Признак c максимальности числа записывается как $(i+j=n)$; он получается на втором шаге как второй слева (т. е. старший значащий) бит числа $[0]^{n+1-i-j}[1]^{i+j}$. На заключительном шаге алгоритма при $c=1$ выбирается исходное число, при $c=0$ результат составляется из фрагмента S исходного числа и вычисленного фрагмента $1\,[0]^{i+1}[1]^{j-1}$. Схема корректно работает и с нулевым входом.

В практической схемотехнике условие $\overline{c}$ может использоваться отдельно как признак валидности результата (для записи его в регистр), тогда финальную операцию можно упростить до $\mathtt{bit}[xy\vee z\overline{y}]$. При этом сложность схемы понизится до $12n+o(n)$.

Оператор множественного выбора. Рассмотрим еще одно обобщение мультиплексора: оператор $\mathtt{SEL}^k_n : [\![n]\!]^k \times \mathbb{B}^n \to \mathbb{B}^k$ реализует выбор k из n информационных булевых переменных по их адресам. Метод Д. Улига (см. [27, 28]) позволяет строить более компактные схемы для множественного выбора в сравнении с простым объединением k независимых схем мультиплексоров. Для демонстрации идеи ограничимся случаем $k=2$.

Справедлива оценка $\mathsf{C}_{\log}(\mathtt{SEL}^2_n) \le 3n+O(n^{2/3})$. Обозначим $m=\lceil \log n \rceil$. Разобьем строку $Y \in \mathbb{B}^n$ на 2^r строк Y_i, $0 \le i < 2^r$, где строка $Y_i =$

$[y_i,\, y_{i+2^r},\, y_{i+2\cdot 2^r}, \ldots]$ включает переменные с индексами, равными i по модулю 2^r. По построению, $|Y_i| = q := \lceil n/2^r \rceil$ (более короткие строки дополним до длины q произвольно). Составим строки

$$\widetilde{Y}_0 = Y_0,\ \ \widetilde{Y}_1 = Y_0 \oplus Y_1,\ \ \ldots,\ \ \widetilde{Y}_i = Y_{i-1} \oplus Y_i,\ \ \ldots,$$
$$\widetilde{Y}_{2^r-1} = Y_{2^r-2} \oplus Y_{2^r-1},\ \ \widetilde{Y}_{2^r} = Y_{2^r-1}, \quad (9)$$

где операции сложения — поразрядные. Заметим, что при любом i

$$\widetilde{Y}_0 \oplus \ldots \oplus \widetilde{Y}_i = Y_i = \widetilde{Y}_{i+1} \oplus \ldots \oplus \widetilde{Y}_{2^r}.$$

Обозначим через $a = [a_1,\, a_0]$ и $b = [b_1,\, b_0]$ адресные входы схемы — в них выделены группы младших r разрядов: $|a_0| = |b_0| = r$.

Таким образом, если $a_0 \le b_0$, то можно определить

$$\mathtt{SEL}^2_n(a, b; Y) = [\mathtt{MUX}_n(a; Y),\, \mathtt{MUX}_n(b; Y)] = [\mathtt{MUX}_q(a_1; Y_{a_0}),\, \mathtt{MUX}_q(b_1; Y_{b_0})]$$

при помощи операторов $\mathtt{MUX}_q(z_i; \widetilde{Y}_i)$, полагая $z_i = a_1$ при $0 \le i \le a_0$ и $z_i = b_1$ при $b_0 < i \le 2^r$. В случае $a_0 > b_0$ переменные a и b меняются местами. В согласии с указанным правилом введем функции-индикаторы

$$\eta^a_i = (i \le a_0 \le b_0) \vee (i > a_0 > b_0), \quad \eta^b_i = (i > b_0 \ge a_0) \vee (i \le b_0 < a_0), \quad (10)$$

выбирающие, использовать подсхему $\mathtt{MUX}_q(z_i; \widetilde{Y}_i)$ для вычисления $\mathtt{MUX}_n(a; Y)$ или для вычисления $\mathtt{MUX}_n(b; Y)$. Окончательно результат определяется по формулам

$$\mathtt{MUX}_n(a; Y) = \bigoplus_{i=0}^{2^r} \eta^a_i\, \mathtt{MUX}_q(z_i; \widetilde{Y}_i), \quad \mathtt{MUX}_n(b; Y) = \bigoplus_{i=0}^{2^r} \eta^b_i\, \mathtt{MUX}_q(z_i; \widetilde{Y}_i),$$
$$z_i = [\eta^a_i]^{m-r} \cdot a_1 \vee [\eta^b_i]^{m-r} \cdot b_1, \quad (11)$$

где операции в последней формуле — поразрядные. Сложность схемы складывается из вычислений, предусмотренных (9), (10), (11), и оценивается как

$$\mathsf{C}_{\log}(\mathtt{SEL}^2_n) \le (2^r - 1)q + (2^r + 1)\mathsf{C}_{\log}(\mathtt{MUX}_q) + O(2^r m) \le$$
$$\le 3 \cdot 2^r q + O(2^r(m + \sqrt{q}) + q).$$

Требуемая оценка получается при выборе $r \approx \log n/3$. Метод практичен уже при $r = 1$.

На самом деле, линейная верхняя оценка сложности $\mathsf{C}_{\log}(\mathtt{SEL}^k_n) = O(n)$ справедлива для всех $k \le n/\log^3 n$, как показано в [17] весьма нетривиальным методом.

Перестановка пары битов. Оператор $\mathtt{EXC}_n : [\![n]\!]^2 \times \mathbb{B}^n \to \mathbb{B}^n$ выполняет перестановку двух бит с заданными адресами в булевой строке длины n. Эта довольно популярная операция (особенно в программировании) обсуждается, например, в [3, §7.1.3].

Опираясь на результат предыдущего пункта, несложно вывести оценку $\mathsf{C}_{\log}(\mathtt{EXC}_n) \leq 7n + O(n^{2/3})$. Решение также использует известный прием обмена содержимым между двумя регистрами, см. [10, Гл. 2]. Вычислим искомую пару битов $[u, v] = \mathtt{SEL}^2_n(a, b; Y)$. Затем при помощи двух демультиплексоров построим строки $\mathtt{DEC}^*_n(a; u \oplus v)$ и $\mathtt{DEC}^*_n(b; u \oplus v)$ с битами $u \oplus v$ в каждой из заданных позиций a, b. Поразрядно сложим эти строки друг с другом и с исходной строкой Y (по модулю 2).

Автор благодарен рецензенту за полезные замечания, в частности, за наблюдения, позволившие уточнить сложность преобразования из унарной кодировки в бинарную, приоритетного шифратора и 2-селектора.

Список литературы

[1] Клосс Б.М., Малышев В.А., “Оценки сложности некоторых классов функций”, *Вестник Моск. университета. Серия 1. Матем. Мех.*, 1965, № 4, 44–51.

[2] Кнут Д.Э., *Искусство программирования. Т.* 2. *Получисленные алгоритмы*, Вильямс, М., 2004, 832 с.

[3] Кнут Д.Э., *Искусство программирования. Т.* 4, *А. Комбинаторные алгоритмы, часть* 1, Диалектика, М., 2020, 960 с.

[4] Митягин Б.С., Садовский Б.Н., “О линейных булевских операторах”, *Доклады АН СССР*, **165**:4 (1965), 773–776.

[5] Офман Ю.П., “Алгоритмическая сложность дискретных функций”, *Доклады АН СССР*, **145**:1 (1962), 48–51.

[6] Редькин Н. П., “О минимальной реализации двоичного сумматора”, *Проблемы кибернетики. Вып.* 38, Наука, М., 1981, 181–216.

[7] Сергеев И.С., “Верхние оценки глубины симметрических булевых функций”, *Вестник Моск. университета. Серия 15. Выч. матем. и киберн.*, 2013, № 4, 39–44.

[8] Сергеев И.С., “О сложности монотонных схем для пороговых симметрических булевых функций”, *Дискретная математика*, **32**:1 (2020), 81–109. DOI: 10.4213/dm1547.

[9] Угрюмов Е.П., *Цифровая схемотехника*, БХВ-Петербург, СПб., 2010, 816 с.

[10] Уоррен Г.С., мл., *Алгоритмические трюки для программистов*, Вильямс, М., 2003, 288 с.

[11] Харрис Д.М., Харрис С.Л., *Цифровая схемотехника и архитектура компьютера*, ДМК Пресс, М., 2017, 792 с.

[12] Чашкин А.В., “О сложности булевых матриц, графов и соответствующих им булевых функций”, *Дискретная математика*, **6**:2 (1994), 43–73.

[13] Чашкин А.В., *Дискретная математика*, Академия, М., 2012, 352 с.

[14] Ahn J.-S., Jeong D.-K., Yang M., "Fast three-dimensional programmable two-selector", *Electronics Letters*, **40**:18 (2004), 1098–1100. DOI: 10.1049/el:20045935

[15] Blelloch G.E., "Prefix sums and their applications", *Synthesis of parallel algorithms*, Morgan Kaufmann, San Francisco, 1993, 35–60

[16] Demenkov E., Kojevnikov A., Kulikov A., Yaroslavtsev G., "New upper bounds on the Boolean circuit complexity of symmetric functions", *Inform. Proc. Letters*, **110**:7 (2010), 264–267. DOI: 10.1016/j.ipl.2010.01.007

[17] Holmgren J., Rothblum R., "Linear-size Boolean circuits for multiselection", *Proc. CCC (Ann Arbor, USA,* 2024*). LIPIcs*, **300**, 2024, 11:1–11:20. DOI: 10.4230/LIPIcs.CCC.2024.11

[18] Klein P., Paterson M.S., "Asymptotically optimal circuit for a storage access function", *IEEE Trans. on Computers*, **C-29**:8 (1980), 737–738. DOI: 10.1109/TC.1980.1675657

[19] Ladner R.E., Fischer M.J., "Parallel prefix computation", *J. ACM*, **27**:4 (1980), 831–838. DOI: 10.1145/322217.322232

[20] Lamagna E.A., Savage J.E., "On the logical complexity of symmetric switching functions in monotone and complete bases", *Tech. report*, Brown Univ., Rhode Island, 1973

[21] Nakano K., Yamagishi Y., "Hardware n choose k counters with applications to the partial exhaustive search", *IEICE Trans. on Information & Systems*, **E88-D**:7 (2005), 1350–1359. DOI: 10.1093/ietisy/e88-d.7.1350

[22] Paterson M.S., Pippenger N., Zwick U., "Optimal carry save networks", *LMS Lecture Notes Series. Boolean function complexity. Vol.* 169, Cambridge Univ. Press, Cambridge, 1992, 174–201. DOI: 10.1017/CBO9780511526633.014

[23] Paul W.J., "A $2.5n$-lower bound on the combinational complexity of Boolean functions", *SIAM J. Comput.*, **6**:3 (1977), 427–443. [Перевод: Пауль В.Й., "Нижняя оценка $2.5n$ для комбинационной сложности булевых функций", *Кибернетический сборник. Вып.* 16, Мир, М., 1979, 23–44.]. DOI: 10.1145/800116.803750

[24] Sergeev I.S., "On the complexity of parallel prefix circuits", *Electr. Colloq. on Comput. Complexity, TR13–041*, 2013

[25] Sklansky J., "Conditional-sum addition logic", *IRE Trans. Electr. Comput*, **EC-9** (1960), 226–231. DOI: 10.1109/TEC.1960.5219822

[26] Stockmeyer L.J., "On the combinational complexity of certain symmetric Boolean functions", *Math. Systems Theory*, **10** (1977), 323–336 [Перевод: Стокмейер Л.Дж., "О комбинационной сложности некоторых симметрических булевых функций", *Кибернетический сборник. Вып.* 16, Мир, М., 1979, 45–61.]. DOI: 10.1007/BF01683282

[27] Uhlig D., "Networks computing Boolean functions for multiple input values", *LMS Lecture Notes Series. Boolean function complexity. Vol.* 169, Cambridge Univ. Press, Cambridge, 1992, 165–173. DOI: 10.1017/CBO9780511526633.013

[28] Wegener I., *The complexity of Boolean functions*, Wiley–Teubner, Stuttgart, 1987, 470 с.

Статья поступила 12 января 2026 г.

Complexity of basic boolean operators for digital circuit design

I. S. Sergeev

This article provides a survey of circuit complexity bounds for basic boolean transforms exploited in digital circuit design and efficient methods for synthesizing such circuits. The exposition covers structurally simple functions and operators, such as counters, adders, encoders, and multiplexors, and excludes more complex algebraic operations with numbers, polynomials, and matrices. Several applications to implementing more specific operations are also discussed.

Keywords: boolean circuits, complexity, depth, parallel circuits, prefix circuits, incrementor, up-down counter, Gray counter, adder, comparator, decoder, multiplexor, encoder, compressor, threshold symmetric functions, priority encoder, unary coding, sorting.

References

[1] Kloss B.M., Malyshev V.A., "Otsenki slozhnosti nekotorykh klassov funktsii [Complexity bounds for some classes of functions]", *Vestnik Mosk. Univ. Ser. 1. Mat. Mekh.*, 1965, № 4, 44–51 (In Russian)

[2] Knuth D.E., *The art of computer programming. Vol.* 2. *Seminumerical algorithms*, Addison-Wesley, Reading, 1997, 762 pp.

[3] Knuth D.E., *The art of computer programming. Vol.* 4*A*. *Combinatorial algorithms, part* 1, Addison-Wesley, Upper Saddle River, 2011, 883 pp.

[4] Mitiagin B.S., Sadovskii B.N., "O lineinykh bulevskikh operatorakh [On linear Boolean operators]", *Dokl. AN SSSR*, **165**:4 (1965), 773–776 (In Russian)

[5] Ofman Y.P., "On the algorithmic complexity of discrete functions", *Soviet Phys. Dokl.*, **7** (1963), 589–591

[6] Red'kin N.P., "O minimal'noi realizatsii dvoichnogo summatora [On the minimal implementation of a binary adder]", *Problemy Kibernetiki. Vol.* 38, Nauka, Moscow, 1981, 181–216 (In Russian)

[7] Sergeev I.S., "Upper bounds on the depth of symmetric Boolean functions", *Moscow Univ. Comput. Math. and Cybern.*, **37**:4 (2013), 195–201. DOI: 10.3103/S0278641913040080

[8] Sergeev I.S., "On the complexity of monotone circuits for threshold symmetric Boolean functions", *Discrete Math. and Appl.*, **31**:5 (2021), 345–366. DOI: 10.1515/dma-2021-0031

[9] Ugryumov E.P., *Tsifrovaya skhemotekhnika* [*Digital circuit design*], BHV-Peterburg, Saint Petersburg, 2010 (In Russian), 816 pp.

[10] Warren H.S., Jr., *Hacker's delight*, Addison-Wesley, Upper Saddle River, 2002, 494 pp.

[11] Harris D.M., Harris S.L., *Digital design and computer architecture*, Morgan Kaufmann, San Francisco, 2012, 712 pp.

[12] Chashkin A.V., "On the complexity of Boolean matrices, graphs and their corresponding Boolean functions", *Discrete Math. and Appl.*, **4**:3 (1994), 229–257. DOI: 10.1515/dma.1994.4.3.229

[13] Chashkin A.V., *Diskretnaya matematika* [*Discrete mathematics*], Akademiya, Moscow, 2012 (In Russian), 352 pp.

[14] Ahn J.-S., Jeong D.-K., Yang M., "Fast three-dimensional programmable two-selector", *Electronics Letters*, **40**:18 (2004), 1098–1100. DOI: 10.1049/el:20045935

[15] Blelloch G.E., "Prefix sums and their applications", *Synthesis of parallel algorithms*, Morgan Kaufmann, San Francisco, 1993, 35–60

[16] Demenkov E., Kojevnikov A., Kulikov A., Yaroslavtsev G., "New upper bounds on the Boolean circuit complexity of symmetric functions", *Inform. Proc. Letters*, **110**:7 (2010), 264–267. DOI: 10.1016/j.ipl.2010.01.007

[17] Holmgren J., Rothblum R., "Linear-size Boolean circuits for multiselection", *Proc. CCC (Ann Arbor, USA,* 2024*). LIPIcs*, **300**, 2024, 11:1–11:20. DOI: 10.4230/LIPIcs.CCC.2024.11

[18] Klein P., Paterson M.S., "Asymptotically optimal circuit for a storage access function", *IEEE Trans. on Computers*, **C-29**:8 (1980), 737–738. DOI: 10.1109/TC.1980.1675657

[19] Ladner R.E., Fischer M.J., "Parallel prefix computation", *J. ACM*, **27**:4 (1980), 831–838. DOI: 10.1145/322217.322232

[20] Lamagna E.A., Savage J.E., "On the logical complexity of symmetric switching functions in monotone and complete bases", *Tech. report*, Brown Univ., Rhode Island, 1973

[21] Nakano K., Yamagishi Y., "Hardware n choose k counters with applications to the partial exhaustive search", *IEICE Trans. on Information & Systems*, **E88-D**:7 (2005), 1350–1359. DOI: 10.1093/ietisy/e88-d.7.1350

[22] Paterson M.S., Pippenger N., Zwick U., "Optimal carry save networks", *LMS Lecture Notes Series. Boolean function complexity. Vol.* 169, Cambridge Univ. Press, Cambridge, 1992, 174–201. DOI: 10.1017/CBO9780511526633.014

[23] Paul W.J., "A $2.5n$-lower bound on the combinational complexity of Boolean functions", *SIAM J. Comput.*, **6**:3 (1977), 427–443. DOI: 10.1145/800116.803750

[24] Sergeev I.S., "On the complexity of parallel prefix circuits", *Electr. Colloq. on Comput. Complexity, TR13–041*, 2013

[25] Sklansky J., “Conditional-sum addition logic”, *IRE Trans. Electr. Comput*, **EC-9** (1960), 226–231. DOI: 10.1109/TEC.1960.5219822

[26] Stockmeyer L.J., “On the combinational complexity of certain symmetric Boolean functions”, *Math. Systems Theory*, **10** (1977), 323–336. DOI: 10.1007/BF01683282

[27] Uhlig D., “Networks computing Boolean functions for multiple input values”, *LMS Lecture Notes Series. Boolean function complexity. Vol.* 169, Cambridge Univ. Press, Cambridge, 1992, 165–173. DOI: 10.1017/CBO9780511526633.013

[28] Wegener I., *The complexity of Boolean functions*, Wiley–Teubner, Stuttgart, 1987, 470 pp.

Received on January 12, 2026

Таблица 1: Оценки сложности базовых операторов

опер. F	нижняя оценка $\mathsf{C}(F)$	верхняя оценка $\mathsf{C}(F)$	верхняя оценка $\mathsf{C}_{\log}(F)$
$\mathtt{PREF}_n$	$n-1$		$2n-\Theta(\log n)$ [5]
$\mathtt{PS}_n$	$2n-3$		$3n-\Theta(\log n)$
$\mathtt{INC}_n$	$2n-2$		$3n-\Theta(\log n)$
$\mathtt{UDC}_n$	—	$3n-3$	$4n-\Theta(\log n)$
$\mathtt{GRC}_n$	—	$4n-7$	$6n-\Theta(\log n)$
$\mathtt{CAR}_n$	$2n-2$		$5n-\Theta(\log n)$
$\mathtt{ADD}_n$	$5n-3$ [6]		$8n-\Theta(\log n)$
$\mathtt{CMP}_n$	—	$4n-3$	$5n-\Theta(\log n)$
$\mathtt{MAX}_n$	—	$6n-3$	$7n-\Theta(\log n)$
$\mathtt{DEC}_n$	$n+\Theta(\sqrt{n})$		
$\mathtt{MUX}_n$	$2n-2$ [23]	$2n+O(\sqrt{n})$ [18]	
$\mathtt{MUX}_n^k$	—	$2kn+O(\sqrt{kn})$	
$\mathtt{CYC}_{k,n}$	—	$3\lceil\log k\rceil n$	
$\mathtt{SFT}_{k,n}$	—	$3\lceil\log k\rceil n-\Theta(k)$	
$\mathtt{ENC}_n$	$2(n-\lceil\log n\rceil-1)$ [12]		
$\mathtt{UN}_n$	—	$2n+O(\sqrt{n})$	
$\mathtt{UN}_n^{-1}$	$n-1$		
$\mathtt{TRN}_n$	—	$3n+O(\sqrt{n})$	
$\mathtt{FOI}_n$	$2n-2$		$3n-\Theta(\log n)$
$\mathtt{PENC}_n$	$2n-\Theta(\log n)$	$2n-3$	$3n-\Theta(\log n)$
$\mathtt{SUM}_n$	$2.5n+\Theta(\log n)$ [20, 26]	$4.5n-\Theta(\log n)$ [16]	$4.5n+o(n)$ [7]
$\mathtt{THR}_n^k$	$2n+\min\{k,\, n-k\}-5$ [26]	$4.5n+O(\log n)$	$4.5n+o(n)$
$\mathtt{BW}_n$	—	$4n-\Theta(\log n)$	$4n+o(n)$
$\mathtt{SORT}_n$	$3n-6$ [1, 20]	$6.5n+O(\sqrt{n})$	$6.5n+o(n)$