УДК 004.384:004.272:004.414.2

Э.И. Ватутин, И.В. Зотов, В.С. Титов

ИСПОЛЬЗОВАНИЕ СХЕМНЫХ ФОРМИРОВАТЕЛЕЙ И ПРЕОБРАЗОВАТЕЛЕЙ ДВОИЧНЫХ ПОСЛЕДОВАТЕЛЬНОСТЕЙ ПРИ ПОСТРОЕНИИ КОМБИНАТОРНО-ЛОГИЧЕСКИХ АКСЕЛЕРАТОРОВ

В статье рассматриваются существующие и предлагаются новые комбинационные схемы для формирования и преобразования двоичных последовательностей (битовых векторов), используемые в составе устройств-акселераторов, обеспечивающих ускоренное решение комбинаторно-логических задач проектирования и программирования вычислительных систем. Дается оценка быстродействия и аппаратной сложности рассматриваемых схем.

E.I. Vatutin, I.V. Zotov, V.S. Titov

USING HARDWIRED BINARY SEQUENCE GENERATORS AND TRANSFORMERS IN THE CONSTRUCTION OF COMBINATORIAL LOGIC ACCELERATORS

In the paper, existing and novel combinatorial circuits for binary sequence (bit vectors) generation and transformation are considered, the circuits being a part of acceleration hardware employed to speed up the search for solutions to combinatorial logic problems that arise in the design and programming of computer systems. The processing speed and hardware complexity of the considered circuits are evaluated.

Одной из основных тенденций развития современной вычислительной техники является повышение ее универсальности. Существующие вычислительные системы позволяют решать весьма широкие классы задач. В то же время известен ряд задач, эффективное решение которых универсальными средствами принципиально невозможно; это так называемые NP-трудные задачи [1]. К их числу относятся (или к ним сводятся) многие задачи проектирования вычислительной техники, например, размещение элементов на печатных платах и трассировка электрических соединений [2]. Подобные задачи распространены и при программировании вычислительных (и управляющих) систем. К ним, в частности, можно отнести размещение программных фрагментов между процессорными модулями [3-5]. Кроме того, задачей такого рода является выбор разбиений алгоритмов управления при их реализации в логических мультиконтроллерах [6]. Достижение высокого качества разбиения требует существенных временных затрат, и обработка алгоритмов реальной размерности (порядка 1000-10000 вершин) может потребовать от нескольких месяцев до нескольких лет машинного времени на универсальных ЭВМ. Решение задач секвенирования (поиска последовательностей био-

полимеров) при анализе ДНК и РНК с применением скрытых моделей Маркова также обладает большой вычислительной сложностью [7]. Ускорение решения задач указанного выше класса достигается переносом наиболее трудоемких операций на аппаратный уровень путем разработки соответствующих устройств-акселераторов, адаптированных к структуре решаемой задачи. Описание подобных устройств можно найти, например, в работах [2, 8-10].

Ядром большинства известных акселераторов являются схемы формирования и преобразования двоичных последовательностей (битовых векторов). От их сложности и быстродействия, в конечном счете, зависит эффективность акселератора в целом. Наиболее распространенные из таких схем обсуждаются в данной статье. Предлагается ряд новых схемных решений. Все рассматриваемые в статье схемы являются комбинационными и допускают следующие способы реализации: 1) построение минимальной ДНФ (или КНФ) для выходов как функций входов (возможно, с учетом некоторых предварительных преобразований); 2) построение схем с использованием алгоритмических особенностей выполняемых ими преобразований; 3) реализацию в базисе ПЛМ.

Схема подсчета бит (СПБ)

Исходными данными для схемы выступает двоичный вектор X. В результате обработки на выходе схемы должно быть сформировано двоичное значение, равное числу бит вектора X, установленных в единицу. На рис. 1–3 приведены таблицы истинности и карты Карно для искомых схем с различной разрядностью вектора X. Видно, что с ростом разрядности вектора X сложность схемы растет нелинейно, однако время задержки получения результата остается постоянным и составляет $3t_0$, где t_0 – время задержки одного логического элемента. Подобным образом можно синтезировать схему для любой разрядности вектора X.

Из общих закономерностей, подмеченных в ходе анализа рис. 1–3, можно отметить лишь формулу для младшего разряда результата

$$f_1 = x_1 \oplus x_2 \oplus \ldots \oplus x_n .$$

Формулы остальных разрядов, по-видимому, выводятся для каждого случая индивидуально.

x_1	x_2	f_2	f_1
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$f_2 = x_1 x_2$

x_1 \ x_2	0	1
0	0	0
1	0	1

$f_1 = x_1\overline{x}_2 \vee \overline{x}_1 x_2 = x_1 \oplus x_2$

x_1 \ x_2	0	1
0	0	1
1	1	0

Рис. 1. Таблица истинности, карты Карно и формулы выходов СПБ при количестве входов n=2

x_1	x_2	x_3	f_2	f_1
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$f_2 = x_1x_3 \vee x_2x_3 \vee x_1x_2$

x_1 \ x_2x_3	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$f_1 = x_1\bar{x}_2\bar{x}_3 \vee \bar{x}_1\bar{x}_2x_3 \vee x_1x_2x_3 \vee \bar{x}_1x_2\bar{x}_3 = x_1 \oplus x_2 \oplus x_3$

x_1 \ x_2x_3	00	01	11	10
0	0	1	0	1
1	1	0	1	0

Рис. 2. Таблица истинности, карты Карно и формулы выходов СПБ при количестве входов n=3

x_1	x_2	x_3	x_4	f_3	f_2	f_1
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	0
0	1	1	1	0	1	1
1	0	0	0	0	0	1
1	0	0	1	0	1	0
1	0	1	0	0	1	0
1	0	1	1	0	1	1
1	1	0	0	0	1	0
1	1	0	1	0	1	1
1	1	1	0	0	1	1
1	1	1	1	1	0	0

$f_3 = x_1x_2x_3x_4$

x_1x_2 \ x_3x_4	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	1	0
10	0	0	0	0

$f_2 = x_1x_2\bar{x}_3 \vee x_1\bar{x}_2x_3 \vee x_1\bar{x}_2x_4 \vee x_1x_3\bar{x}_4 \vee \bar{x}_1x_3x_4 \vee x_2\bar{x}_3x_4$

x_1x_2 \ x_3x_4	00	01	11	10
00	0	0	1	0
01	0	1	1	1
11	1	1	0	1
10	0	1	1	1

$f_1 = x_1 \oplus x_2 \oplus x_3 \oplus x_4$

x_1x_2 \ x_3x_4	00	01	11	10
00	0	1	0	1
01	1	0	1	0
11	0	1	0	1
10	1	0	1	0

Рис. 3. Таблица истинности, карты Карно и формулы выходов СПБ при количестве входов n=4

В [2] синтезируемая схема подсчета бит именуется «цифровым компрессором», а в [11] предлагается ее пирамидальный вариант (рис. 4).

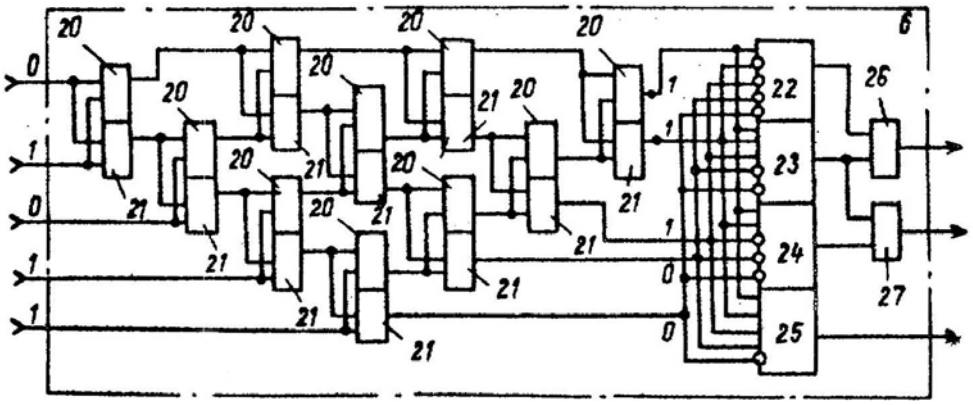

Рис. 4. Пирамидальный вариант схемы подсчета бит (номером 20 обозначены элементы ИЛИ, 21 – И, 22–25 – элементы запрета, 26 и 27 – ИЛИ)

Принцип действия схемы на рис. 4 достаточно прост. Пирамидальная часть схемы, именуемая в дальнейшем схемой упорядочения двоичного вектора (СУДВ), образована элементами ИЛИ 20 и И 21. Она производит упорядочение (компандирование [11]) исходного вектора таким образом, что в одной его части оказываются единичные значения, а в другой – нулевые (например, двоичный вектор «01101011011» будет преобразован в вектор «11111110000»). Это обеспечивается благодаря парам элементов ИЛИ и И (рис. 5), фактически производящим обмен соседних разрядов вектора местами в случае, если $x_1 < x_2$. Указанные пары элементов ИЛИ и И образуют пирамидальную структуру схемы, которая обеспечивает попарное сравнение и, при необходимости, обмен всех бит исходного вектора. В результате происходит перемещение единичных значений в один конец вектора, а нулевых – в другой. Похожий принцип попарных сравнений соседних элементов используется в широко известном алгоритме пузырьковой сортировки массивов. Подобная схема используется и при реализации приоритетных шифраторов.

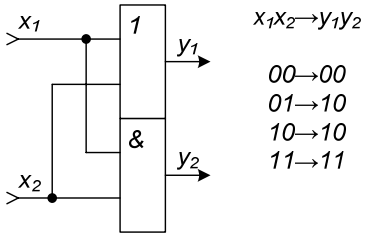

Рис. 5. Элементы ИЛИ и И, производящие упорядочение пары битовых значений

Далее обработкой полученного вектора занимается комбинационная часть схемы, образованная элементами 22–27 (см. рис. 4). За счет того, что в таблицах истинности синтезируемых функций результата встречается большое количество запрещенных комбинаций (см. рис. 6–8, символ «*»), которые не могут появиться на входах комбинационной части схемы благодаря схеме СУДВ, в картах Карно присутствует большое количество звездочек. Это уменьшает число термов в синтезируемых функциях и, соответственно, аппаратную сложность комбинационной

части схемы по сравнению с приведенным выше вариантом реализации «в лоб» (рис. 1–3).

Время задержки получения результата в пирамидальном варианте схемы подсчета бит с комбинационной частью зависит от размерности входного вектора и составляет величину $(n+2)t_0$.

Для каждого значения размерности входного вектора *n* результирующие формулы выходов комбинационной части схемы, по-видимому, индивидуальны. Схема СУДВ включает в себя

$$2\cdot\big((n-1)+(n-2)+...+1\big)=n(n-1)$$

эквивалентных вентилей.

Таким образом, пирамидальный вариант схемы подсчета бит [11] характеризуется большим временем задержки получения результата при меньших аппаратных затратах (см. табл. 1). С ростом *n* соотношение рассмотренных выше вариантов реализации схем по сложности, по-видимому, сохраняется.

Модификацией указанного выше способа реализации схемы подсчета бит является использование вместо комбинационной части схемы выделения старшей единицы (сокр. СВСЕ, см. ниже) и шифратора. Структурная схема подобного преобразования представлена на рис. 9 (в составе схемы СВСЕ можно обойтись без цепочки элементов ИЛИ, т.к. двоичный вектор на ее входе уже представлен в виде «11…100…0» и сочетание входных сигналов «10» на входах элементов запрета может встретиться только однократно). На выходе схемы СВСЕ в унитарном коде формируется вектор из нулей и единицы в *i*-й позиции, где *i* – искомое число единичных бит в исходном векторе. Шифратор переводит полученное значение из унитарного кода в двоичный код.

Быстродействие подобного варианта составляет $(n+4)t_0$ (при условии, что быстродействие шифратора составляет $2t_0$ и не зависит от *n*). При несколько большей аппаратной сложности этого варианта реализации (по сравнению с предыдущим) и большем времени задержки структурная схема (см. рис. 9) обладает регулярностью, что позволяет синтезировать функциональную схему для произвольного значения *n* без дополнительных затрат (например, минимизации булевых функций выходов, индивидуальных для каждого значения *n*).

x_1	x_2	f_2	f_1
0	0	0	0
0	1	*	*
1	0	0	1
1	1	1	0

$f_2 = x_2$

x_1 \ x_2	0	1
0	0	*
1	0	1

$f_1 = x_1\overline{x}_2$

x_1 \ x_2	0	1
0	0	*
1	1	0

Рис. 6. Таблица истинности, карты Карно и формулы выходов комбинационной части СПБ при количестве входов n=2

x_1	x_2	x_3	f_2	f_1
0	0	0	0	0
0	0	1	*	*
0	1	0	*	*
0	1	1	*	*
1	0	0	0	1
1	0	1	*	*
1	1	0	1	0
1	1	1	1	1

$f_2 = x_2$

x_1 \ x_2x_3	00	01	11	10
0	0	*	*	*
1	0	*	1	1

$f_1 = x_3 \vee x_1\overline{x}_2 = x_1 \oplus x_2 \oplus x_3$

x_1 \ x_2x_3	00	01	11	10
0	0	*	*	*
1	1	*	1	0

Рис. 7. Таблица истинности, карты Карно и формулы выходов комбинационной части СПБ при количестве входов n=3

x_1	x_2	x_3	x_4	f_3	f_2	f_1
0	0	0	0	0	0	0
0	0	0	1	*	*	*
0	0	1	0	*	*	*
0	0	1	1	*	*	*
0	1	0	0	*	*	*
0	1	0	1	*	*	*
0	1	1	0	*	*	*
0	1	1	1	*	*	*
1	0	0	0	0	0	1
1	0	0	1	*	*	*
1	0	1	0	*	*	*
1	0	1	1	*	*	*
1	1	0	0	0	1	0
1	1	0	1	*	*	*
1	1	1	0	0	1	1
1	1	1	1	1	0	0

$f_3 = x_4$

x_1x_2 \ x_3x_4	00	01	11	10
00	0	*	*	*
01	*	*	*	*
11	0	*	1	0
10	0	*	*	*

$f_2 = x_2\overline{x}_4$

x_1x_2 \ x_3x_4	00	01	11	10
00	0	*	*	*
01	*	*	*	*
11	1	*	0	1
10	0	*	*	*

$f_1 = x_1\overline{x}_2 \vee x_3\overline{x}_4 = x_1 \oplus x_2 \oplus x_3 \oplus x_4$

x_1x_2 \ x_3x_4	00	01	11	10
00	0	*	*	*
01	*	*	*	*
11	0	*	0	1
10	1	*	*	*

Рис. 8. Таблица истинности, карты Карно и формулы выходов комбинационной части СПБ при количестве входов n=4

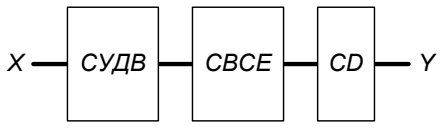

Рис. 9. Вариант схемы подсчета бит с использованием схемы выделения старшей единицы и шифратора

Также возможен вариант построения схемы подсчета бит с использованием пирамиды сумматоров (рис. 10). Похожим способом реализуется подсчет бит

двоичного вектора при программной реализации указанной операции с использованием векторных расширений современных процессоров [12].

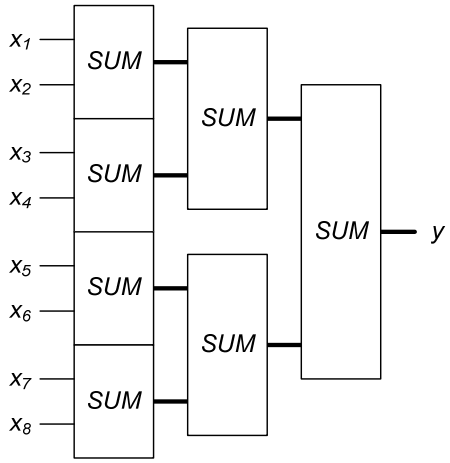

Рис. 10. Вариант схемы подсчета бит с использованием пирамиды сумматоров для случая n=8

Сумматоры первого яруса (на рис. 10 расположены слева) имеют разрядность входов 1 бит, второго яруса – два бита, третьего – три бита и т.д. Время задержки получения результата $t_{SUM\,2} + t_{SUM\,3} + ... + t_{SUM\,\log_2 n} \approx \log_2 n \cdot t_{SUM}$ и аппаратная сложность $\frac{n}{2} R_{SUM\,2} + \frac{n}{4} R_{SUM\,3} + \frac{n}{8} R_{SUM\,4} + ... + R_{SUM\,\log_2 n}$ схемы полностью определяются параметрами используемых при этом сумматоров.

Табл. 1. Сравнение аппаратной сложности вариантов реализации схемы подсчета бит в числе эквивалентных вентилей

n	Комбинационная схема	Совокупность пирамидальной и комбинационной схем	Совокупность пирамидальной схемы, схемы СВСЕ и шифратора	Пирамида сумматоров
2	6	2+2=4	2+2+1=5	~
3	19	6+3=9	6+4+3=13	~
4	55	12+6=18	12+6+3=21	~

Схема маскировки неиспользуемых позиций (СМНП)

Исходными данными для схемы СМНП является двоичное число *X*, определяющее количество единичных бит в результирующем двоичном векторе на выходе схемы, причем единичные значение группируются в начальных (младших) позициях результирующего вектора. Например, если на вход схемы подан вектор «000101» (5 в десятичной форме), то на ее выходе должен быть сформирован вектор «1111100…0». Разрядность результирующего вектора *n* определяется потреб-

ностями более сложных схем, использующих схему СМНП в качестве составной части. Свое название схема получила из-за специфики применения: при необходимости можно «маскировать» (сбросить в ноль) часть бит произвольного двоичного вектора *Y* путем поразрядной конъюнкции его элементов y_i с выходами схемы СМНП f_i.

Обобщенный вариант реализации схемы, в полной мере отражающий логику ее работы, приведен на рис. 11.

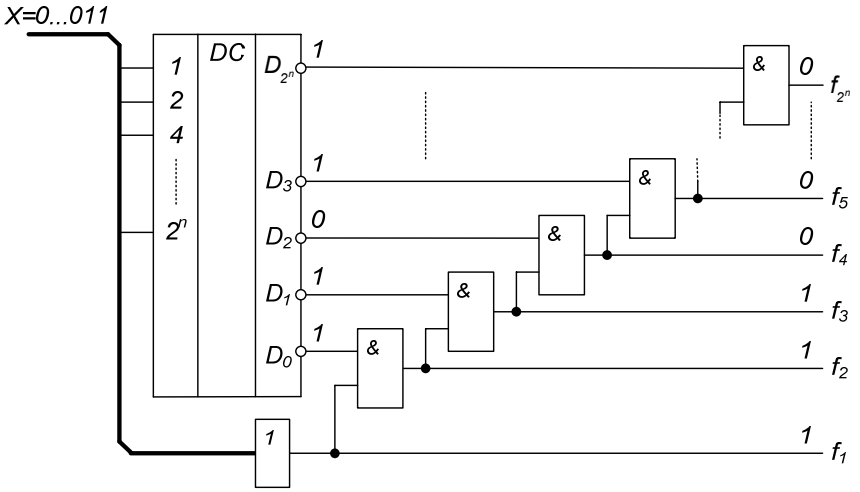

Рис. 11. Обобщенный вариант реализации схемы маскировки неиспользуемых позиций

Соединенные в цепочку элементы И обеспечивают на выходе *i*-го элемента единичное значение только в том случае, если значение текущего (*i*-го) и всех предыдущих (младших) выходов дешифратора равны единице. Нулевое значение на *i*-м выходе дешифратора обеспечивает нулевое значение на выходе *i*-го и всех последующих элементов И. Для того чтобы число единичных бит было равно десятичному представлению *m* двоичного вектора *X*, а не $m-1$, в схему введен элемент ИЛИ, выход которого является младшим разрядом выходного вектора. Нулевое значение на выходе элемента ИЛИ возможно только в том случае, если все разряды вектора *X* равны нулю, т.е. $n=0$. Во всех остальных случаях значение на выходе элемента ИЛИ равно единице.

Рассмотренный вариант схемы СМНП является общим, требует $X_{DC}+2n-1$ эквивалентных вентилей (здесь X_{DC} – аппаратная сложность дешифратора, зависящая от его внутренней структуры) и характеризуется временем задержки получения результата $t_{DC}+(n+1)t_0$, зависящим от числа разрядов выходного вектора (t_{DC} – время формирования сигнала на выходе дешифратора).

При конкретном значении *n* схема допускает упрощение, которое можно получить с использованием карт Карно, однако полученные схемы не обладают свойством регулярности. Пример подобного упрощения для случая $n=8$ приве-

ден на рис. 12 (ввиду ограниченного объема статьи карты Карно для функций выходов схемы на рисунке не приведены).

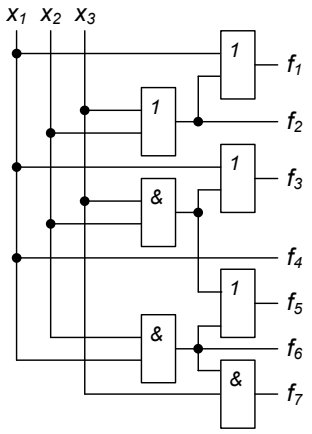

Рис. 12. Пример реализации схемы СМНП в виде МДНФ значений выходного вектора

Приведенная на рис. 12 схема обладает меньшей аппаратной сложностью и характеризуется временем задержки получения результата $2t_0$, не зависящим от n.

В случае необходимости синтеза вектора, в котором набор из m единиц должен быть расположен в конечных (старших) позициях, схемы, приведенные на рис. 11 и 12, остаются прежними, производится лишь перенумерация выходов: первый выход становится последним, второй – предпоследним и т.д.

По-сути СМНП реализует функцию, обратную по отношению к преобразованию, выполняемому рассмотренной выше совокупностью схемы СВСЕ и шифратора.

Схема сравнения битовых векторов (ССБВ)

Исходными данными для этой схемы является пара битовых векторов X и Y, являющихся представлением двух множеств. Результатом работы схемы должны быть двоичные значения признаков полного δ^+ и частичного δ^- совпадения векторов. Очевидно, что признак δ^+ должен получить единичное значение только в случае побитового совпадения векторов:

$$\delta^+ = \underset{i=1,\,n}{\&}\left(x_i \sim y_i\right) = \underset{i=1,\,n}{\&}\left(x_i y_i \vee \overline{x}_i \overline{y}_i\right).$$

Согласно предназначению схемы признак δ^- должен быть установлен в случае, если $X \subset Y$. Другими словами, значение признака δ^- должно быть нулевым, только если в составе векторов на некоторой позиции i $x_i = 1$, а $y_i = 0$ (i-й элемент присутствует в X и не присутствует в Y). Следовательно

$$\delta^- = \underset{i=1,\,n}{\&} f\left(x_i, y_i\right) = \underset{i=1,\,n}{\&} \overline{x_i \overline{y}_i}\,.$$

Использованная в формуле функция синтезирована на основе приведенных выше рассуждений так, как показано на рис. 13.

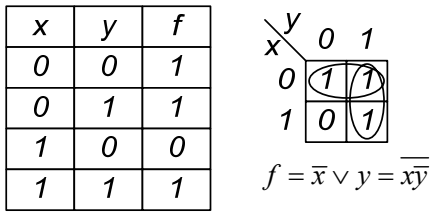

Рис. 13. Таблица истинности и карта Карно для функции δ^-

Формулы, подобные приведенным выше, могут быть использованы и при других операциях над множествами, представленными в виде битовых векторов [13].

На основании полученных выше формул можно синтезировать искомую схему (рис. 14).

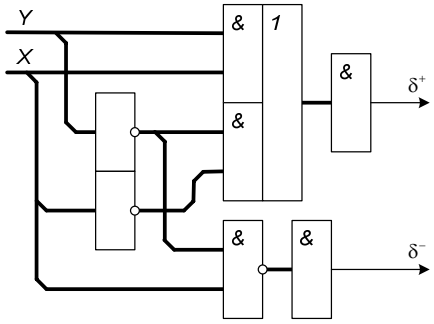

Рис. 14. Схема сравнения битовых векторов

Ее аппаратная сложность составляет $8n-2$ эквивалентных вентилей (n в данном случае – разрядность входных векторов), а время задержки появления результата составляет $\left(3+\lceil \log_2 n \rceil\right)t_0$.

Схемы выделения заданных граничных значений

При выполнении некоторых действий возникает необходимости в нахождении в заданном двоичном векторе самого младшего или самого старшего бита, значение которого равно заданному. Примеры подобных действий приведены в табл. 2.

Таблица 2. Примеры выделения граничных значений

Действие	Исходный вектор X	Результат Y
Выделение младшего нуля	11**0**0101011101010	0010 (2)
Выделение старшего нуля	0011011111**0**11111	1010 (10)
Выделение младшей единицы	000**1**001101111010	0011 (3)
Выделение старшей единицы	1011101111**1**0000	1011 (11)

Комбинационные схемы, выполняющие подобные операции, далее именуются как схема выделения младшего нуля (СВМН), схема выделения старшего нуля (СВСН), схема выделения младшей единицы (СВМЕ) и схема выделения старшей единицы (СВСЕ).

Любая из перечисленных схем может быть синтезирована из минимальной ДНФ, однако (с учетом большого числа входов) минимизация получаемых в ходе анализа таблиц истинности СДНФ с использованием карт Карно напрямую затруднительна, а результирующие формулы и схемы достаточно громоздки. Исходя из этого (не исключая возможности синтеза минимальной ДНФ для выходов схемы) рассмотрим обобщенный способ синтеза схем на примере схемы СВМН (рис. 15).

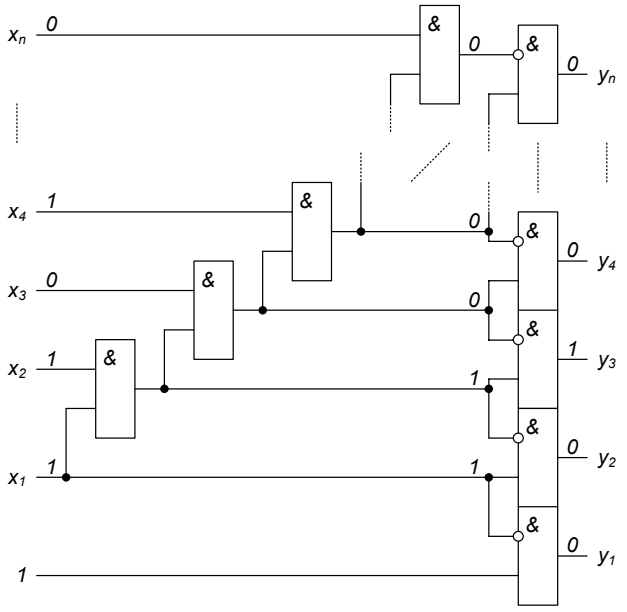

Рис. 15. Обобщенная схема выделение младшего нуля

Элементы И, объединенные в цепочку, преобразуют исходный двоичный вектор в вектор вида «11…100…0», а стоящие следом на ними элементы запрета обеспечивают единицу на выходе того элемента, на входе которого присутствует комбинация значений «10». Т.к. в получаемом таким способом выходном векторе присутствует на один бит меньше, чем в исходном, в качестве младшего разряда (x_0) к исходному вектору добавляется единичный бит. При этом следует отметить, что элемент запрета, формирующий значение младшего разряда выхода, изображен на рис. 15 исключительно из соображений наглядности и общности: при синтезе реальной схемы он может быть заменен инвертором младшего разряда, т.к. $\overline{x}_1 \,\&\, 1 = \overline{x}_1$.

Схема, представленная на рис. 15, характеризуется аппаратной сложностью $2n-1$ эквивалентных вентилей, а время задержки получения результата, составляющее nt_0, линейно зависит от разрядности входного вектора.

СВСН легко получается из рассмотренной выше СВМН путем перенумерации входов: первый выход становится последним, второй – предпоследним и т.д.

СВМЕ может быть синтезирована двумя способами. Первый из них заключается в инвертировании при помощи блока инверторов исходного вектора (включая добавленный младший разряд x_0 в случае использования схемы, представленной на рис. 15) с последующим применением СВМН. Основным недостатком этого способа является увеличение времени формирования результирующего вектора на величину t_0. Второй способ сводится к синтезу самостоятельной схемы (см. рис. 16), принцип работы которой в целом напоминает принцип работы СВМН (рис. 15).

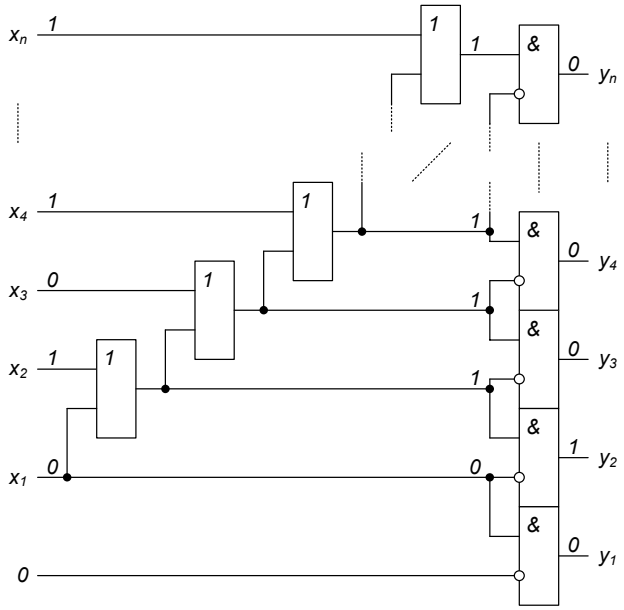

Рис. 16. Схема выделения младшей единицы

На выходах элементов ИЛИ, объединенных в цепочку, исходный вектор преобразуется в вектор вида «00…011…1». Далее на выходе элемента запрета, на вход которого поданы значения «01», формируется единичное значение выходного вектора. На выходах всех остальных элементов запрета формируются нулевые значения. Как и в случае схемы на рис. 15, элемент запрета, на выходе которого формируется значение младшего бита результата, приведен на рис. 16 из соображений наглядности и общности изображения схемы. При синтезе принципиальной схемы он может быть заменен значением младшего бита входного вектора, т.к. $y_1 = x_1 \,\&\, \overline{0} = x_1$.

СВСЕ, аналогично СВСН, может быть получена из приведенной на рис. 16 схемы путем перенумерации входов. Аппаратная сложность схемы на рис. 16 составляет $2n-2$ эквивалентных вентилей, а ее время задержки формирования выходного вектора – nt_0.

Библиографический список

1. Хопкрофт, Дж., Введение в теорию автоматов, языков и вычислений, 2-е изд.: Пер. с англ. [Текст] / Дж. Хопкрофт, Р. Мотвани, Дж. Ульман. М.: «Вильямс», 2002. 528 с.
2. Курейчик, В.М., Комбинаторные аппаратные модели и алгоритмы в САПР [Текст] / В.М. Курейчик, В.М. Глушань, Л.И. Щербаков. М.: «Радио и связь», 1990. 216 с.
3. Lee, C.-H. Optimal task assignment in linear array networks [Текст] / C.-H Lee, D. Lee, M. Kim // IEEE Trans. Comput. 1992. Vol. 41, № 7. P. 877–880.
4. Wu, S.S. Heuristic algorithms for task assignment and scheduling in a processor network [Текст] / S.S. Wu, D. Sweeting // Parallel Comput. 1994. № 20. P. 1–14.
5. Kim, J. Replicated process allocation for load distribution in fault-tolerant multi-computers [Текст] / Jong Kim; Heejo Lee; Sunggu Lee // IEEE Transactions on Computers. Volume 46, Issue 4, 1997. P. 499–505.
6. Зотов, И.В. Архитектура и синтез параллельных логических мультимикроконтроллеров: учебное пособие: в 2 ч. [Текст] / И.В. Зотов, В.С. Титов и др. Курск: КурскГТУ, 2006. 359 с.
7. http://www.ixbt.com/news/all/index.shtml?10/26/71
8. Ватутин, Э.И. Аппаратная модель для определения минимального числа блоков при декомпозиции параллельных алгоритмов логического управления [Текст] / Э.И. Ватутин, И.В. Зотов // Известия вузов. Приборостроение. 2008. Т. 51, № 2. С. 39–43.
9. Ватутин Э.И. Однородная среда электронной модели дерева для аппаратно-ориентированной обработки R-выражений [Текст] / Э.И. Ватутин // Оптико-электронные приборы и устройства в системах распознавания образов, обработки изображений и символьной информации (Распознавание – 2008). Ч. 1. Курск: КурскГТУ, 2008. С. 90–92.
10. Борзов, Д.Б. К задаче субоптимального разбиения параллельных алгоритмов [Текст] / Д.Б. Борзов, Э.И. Ватутин, И.В. Зотов, В.С. Титов // Известия вузов. Приборостроение. Вып. 12, 2004. С. 34–39.
11. Авторское свидетельство СССР № 1273941, МКИ^3 GO6F 15/20. Устройство для разбиения графа на подграфы / В.М. Глушань, Л.И. Щербаков, И.П. Левин. Опубл. 1986, Бюл. № 44.
12. AMD Athlon Processor x86 Code Optimization Guide. Order Number #22007 // http://developer.amd.com, 2000. 320 p.
13. Ватутин, Э.И. Оптимизация обработки множеств [Текст] / Э.И. Ватутин // Медико-экологические информационные технологии 2005. Курск: КурскГТУ, 2005. С. 145–147.

Работа выполнена при поддержке гранта Президента РФ для молодых ученых МК-3073.2007.8.