

УДК 681.142.1.01

СИНТЕЗ СХЕМ С МИНИМАЛЬНОЙ ГЛУБИНОЙ ПРЕОБРАЗОВАНИЯ МЕТОДАМИ СОКРАЩЕННОГО ПЕРЕБОРА

В.П.Битюцкий, Н.В.Закурдаев, Я.В.Ковалин

В настоящей работе рассматриваются алгоритмы сокращения T -перебора [1] при построении комбинационных схем и приводятся некоторые результаты использования этих алгоритмов. Все алгоритмы были запрограммированы на языке ЛЯПАС и реализованы на машине М-20.

1. Метод пробных построений (алгоритм В)

Под пробным построением для заданного выходного списка будем понимать любую схему, реализующую данный выходной список. Пусть пробные построения для функций K -ого уровня таковы, что на $(K-1)$ -ом уровне строятся все образующие списки, а на всех более низких уровнях используются первые образующие списки. В результате будет выполнено столько пробных построений, сколько образующих списков можно построить на $(K-1)$ -ом уровне.

На дереве синтеза этим построениям соответствуют ветви, объединяющие кружочки, обведенные жирной линией на рис. 1. Оценим пробные построения числом элементов схемы и отметим штриховкой на $(K-1)$ -ом уровне те списки, которым соответствует минимальное число элементов. Дальнейший синтез продолжается по отмеченным направлениям. Для этой цели выбирается на $(K-1)$ -ом уровне первый слева отмеченный штриховкой образующий список, для которого вновь выполняются все пробные построения. Для дальнейшего синтеза отбирается первый слева образующий список на $(K-2)$ -ом уровне, соответствующий новому пробному построению с минимальным числом элементов. Этот процесс продолжается вплоть до нулевого уровня. Полученной при этом схеме на рис. 1 соответствует первая слева ветвь, отмеченная жирной линией. Этот

заканчивается первый цикл синтеза.

В h -ом цикле на каждом уровне выбирается не первый, а h -й слева образующий список, соответствующий очередному пробному построению с минимальным числом элементов. Если при этом на каком-либо уровне число списков, соответствующих пробному построению с минимальным числом элементов, меньше h , то к дальнейшему синтезу принимается последний список. На рис. 1 жирными линиями обозначены ветви синтеза для различных h .

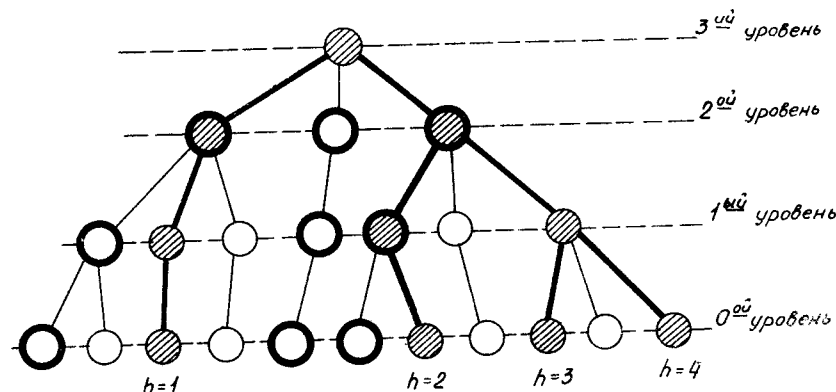


Рис. 1.

Процесс синтеза может начинаться с любого h и заканчивается после выполнения заданного числа циклов.

Описанный алгоритм обозначим буквой B . Алгоритм, в котором с каждой образующей функцией используются парные, находящиеся на минимально возможном уровне, будем обозначать B_1 . Такой алгоритм дает дополнительное сужение перебора.

Еще более узкий перебор можно получить, если в алгоритме B_1 использовать лишь первые парные функции (алгоритм B_2).

В алгоритмах B_1 и B_2 строятся образующие пары со всеми возможными образующими функциями.

Для синтеза схем с максимальными связями отбираются лишь те образующие, для которых парные функции находятся на минимально возможном уровне. Алгоритмы синтеза схем с максимальными связями будем обозначать штриховкой (например B'_1). Полученные схемы оформляются в виде таблицы выходов (T_v) и таблицы связей (T_c).

Таблица выходов представляет собой упорядоченный список функ-

Таблица выходов

Номер уровня	Функция
5	I5I
4	022
4	053
4	204
3	110
3	245
3	304
2	012
2	067
2	I20
1	210
1	240
1	300

Таблица связей

Образующая	Парная
022	204
110	245
304	I20
110	063
067	240
I20	012
012	063
240	I25
210	300
240	017
063	I25
017	I25
017	063

ций, реализованных на каждом уровне с указанием номера уровня.

Таблица связей содержит образующие пары, соответствующие функциям таблицы выходов. Схема, соответствующая данным таблицам выходов и связей, изображена на рис. 2.

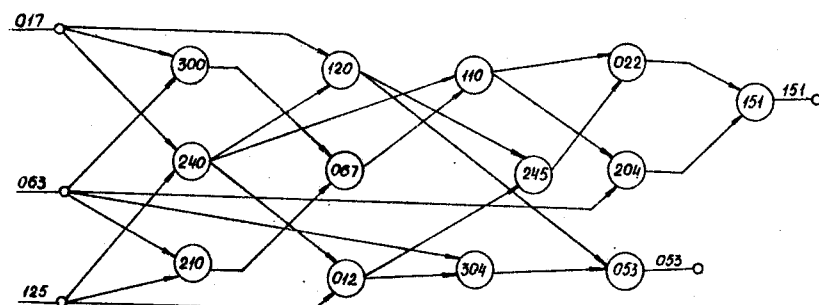


Рис. 2.

На рис. 3 приведены функции распределения схем по числу элементов для одноразрядного сумматора с переносом, полученные рассмотренным методом. Здесь π_0 - общее число просмотренных схем, n - число схем из m элементов, а в скобках указано время перебора схем по одному n . Схемы синтезировались на двухвыходовых элементах "ИЛИ-НЕ". Время, необходимое на построение одной схемы по алгоритмам B , B_1 и B_2 , равно одной секунде, по алгоритмам B'_1 и B'_2 - примерно шести секундам. Несмотря на увеличение времени построения схем с максимальными связями, алгоритмы B'_1 и B'_2 требуют значительно меньшего перебора и позволяют найти за более короткое время схему, по числу элементов близкую к минимальной.

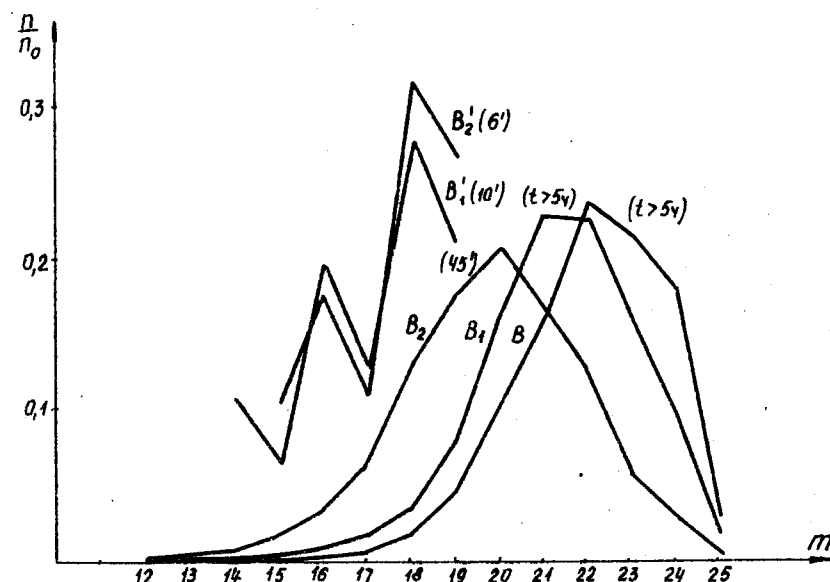


Рис. 3.

Из алгоритмов B , B_1 и B_2 предпочтение следует отдать алгоритму B_2 . При значительном сокращении времени перебора он позволяет найти схему не хуже, чем по другим алгоритмам.

2. Синтез схем с минимальным расширением (алгоритм С)

Пусть число элементов на k -ом уровне логической схемы равно π_k . Тогда

$$\rho_{k-1} = \pi_{k-1} - \pi_k$$

назовем расширением логической схемы (или образующего списка) на $(k-1)$ -м уровне.

Построение логической схемы начинается с формирования всех образующих списков для функций, реализуемых на самом старшем уровне (k). На рис. 4 образующие списки, имеющие минимальное расширение, отмечены штриховкой. Дальнейшие построения проводятся лишь по тем ветвям дерева синтеза, которым соответствуют

на всех уровнях образующие списки с минимальным p_{k-1} . С этой целью выбирается первый слева образующий список $(k-1)$ -го уровня, отмеченный штриховкой, и для него на $(k-2)$ -м уровне строятся все образующие списки. Для дальнейшего синтеза из образующих списков $(k-2)$ -го уровня выбирается первый список с минимальным расширением. Аналогичный процесс продолжается вплоть до нулевого уровня. Полученной при этом схеме на рис. 4 будет соответствовать первая слева ветвь, отмеченная жирной линией. Этим заканчивается первый цикл синтеза. Во втором цикле к дальнейшему синтезу принимаются на каждом уровне только вторые списки с минимальным расширением, в третьем цикле — третьи и т.д. Обозначим номер цикла через h . Тогда на каждом уровне к дальнейшему синтезу принимается h -й список с минимальным расширением. Если на каком-либо уровне число списков с минимальным расширением меньше, чем h , то к дальнейшему синтезу принимается последний список. На рис. 4 жирными линиями отмечены ветви синтеза для различных h . Процесс синтеза можно начинать с любого цикла. Заканчивается синтез после выполнения заданного числа циклов.

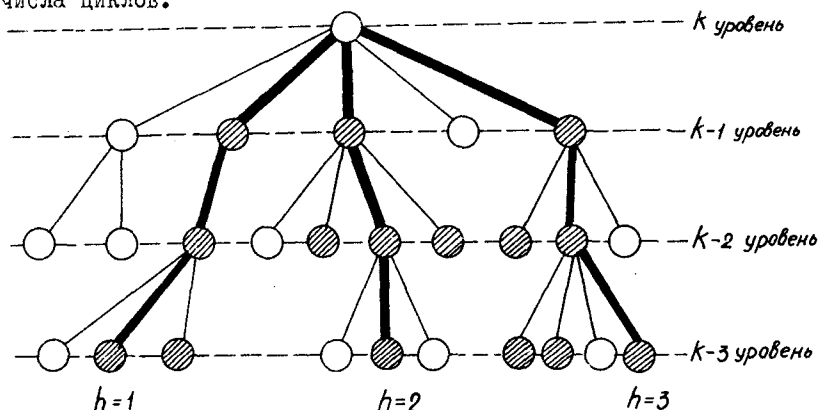


Рис. 4.

Описанный алгоритм будем обозначать буквой C . Алгоритм, в котором при формировании образующего списка для каждой образующей функции используются парные функции, находящиеся на минимально возможном уровне, будем обозначать через C' . Такой алгоритм дает дополнительное сужение перебора. Еще более узкий перебор получается при формировании образующих списков лишь из тех образующих пар, которые содержат первые парные функции с минимально возможного уровня (алгоритм C_2).

Для сравнения с алгоритмом C были построены алгоритмы E и

$C-E$. Алгоритм E на каждом уровне для дальнейшего синтеза принимает образующие списки, содержащие минимальное число тех элементов, которые не использовались в построенной уже части схемы. Алгоритм $C-E$ выбирает на каждом уровне из списков с минимальным расширением только те, у которых число элементов, не используемых ранее, минимально.

В описанные алгоритмы условия максимальных связей вводятся тем же способом, как и в алгоритмы синтеза схем методом пробных построений.

На рисунке 5 приведены функции распределения для множества логических схем одноразрядного сумматора, полученных в результате работы этих алгоритмов. Здесь же приведены затраты времени на каждый алгоритм. Синтез проводился на двухходовых элементах "ИЛИ-НЕ".

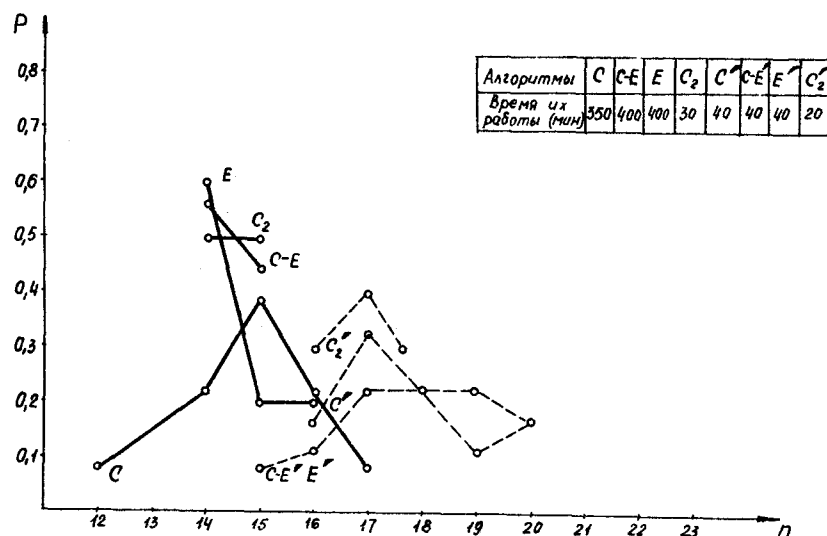


Рис. 5.

Полученные результаты позволяют сделать следующие выводы:

1) Различные модернизации алгоритма C ($C-E$, E , C_2) не дают существенных изменений в качестве получаемых схем. Алгоритм C_2 требует наименьшей затраты машинного времени.

2) Введение максимальных связей при построении схем (алгоритмы C' , $C-E'$, E' , C_2') ухудшает качество получаемых схем. Очевидно, требования минимального расширения и максимальных связей противоречивы.

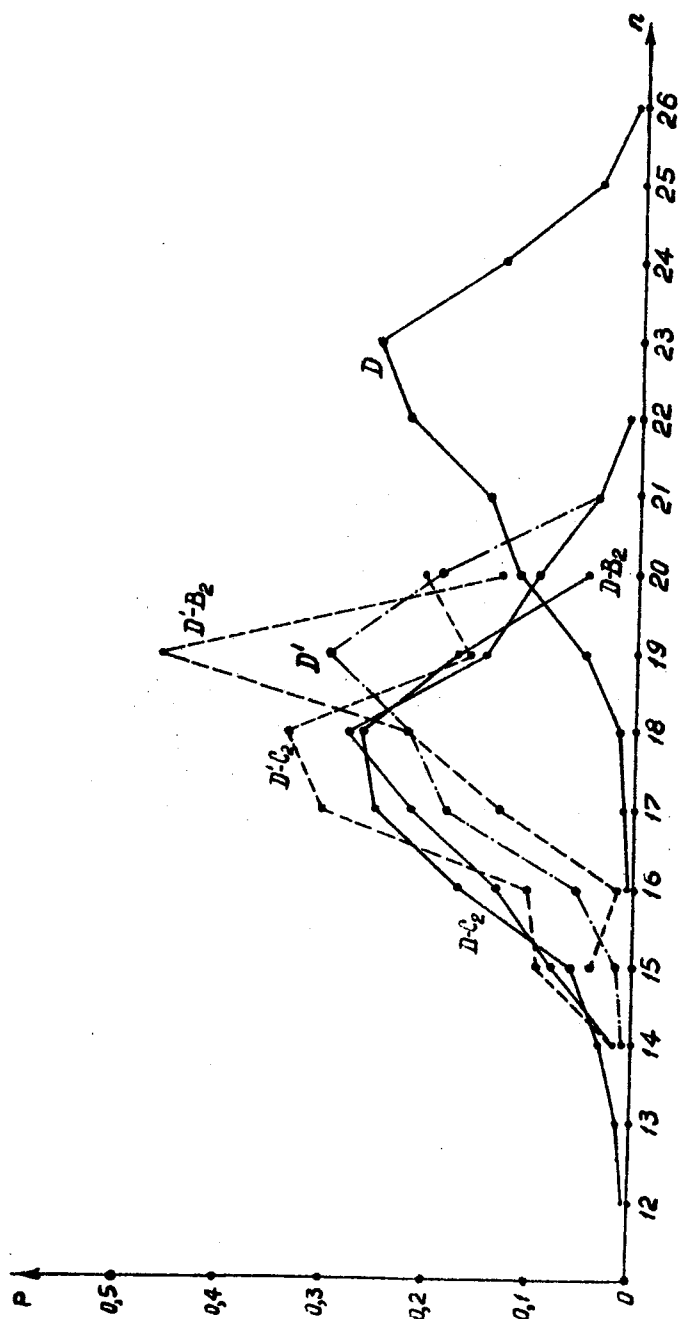


Рис. 6.

3. Синтез схем со случайным способом формирования образующих списков (алгоритм D).

Случайный способ формирования образующих списков заключается в том, что для каждой функции K -го уровня строится матрица образующих функций на $(K-1)$ -м уровне, из которой равновероятным способом выбирается образующая функция. По функции K -го уровня и выбранной образующей строится матрица парных функций. Из нее также равновероятным способом выбирается парная функция. Полученную таким образом пару будем называть случайной образующей парой. Совокупность случайных образующих пар составляет случайный образующий список на $(K-1)$ -м уровне. Формированием случайных образующих списков на каждом уровне строится логическая схема, которую будем называть случайным построением.

Алгоритм D формирует множество случайных построений и работает до построения заданного числа схем.

Алгоритм $D-B_2$ выполняет синтез методом пробных построений со случайным способом формирования списка.

Алгоритм $D-C_2$ выполняет синтез схем с минимальным расширением при случайном способе формирования образующих списков.

Как и в алгоритмах B_2 , C_2 , в алгоритмах $D-B_2$ и $D-C_2$ в матрице парных функций оставляются лишь те, которые находятся на минимально возможном уровне.

Алгоритмы D' , $D'-B_2$ и $D'-C_2$ отличаются тем, что в формировании случайных образующих пар участвуют лишь те пары, которые имеют максимальные связи.

С помощью указанных алгоритмов были построены схемы одноразрядного сумматора из двухвходовых элементов "ИЛИ-НЕ".

Функции распределений вероятности P построения схемы из 72 элементов приведены на рис. 6. Из графиков видно, что введение выбора парных функций с минимально возможного уровня (алгоритмы $D-B_2$ и $D-C_2$) смещает функцию распределения в сторону меньшего числа элементов. Наиболее эффективными являются алгоритмы, строящие схемы с максимальными связями (D' , $D'-B_2$, $D'-C_2$), причем наиболее простой из них (D') дает те же результаты, что и алгоритмы с упорядоченным спуском.

Л и т е р а т у р а

И. В. П. Чистов. Синтез комбинационных схем с минимальной глубиной преобразования. Настоящий сборник, стр. 55-71.

Поступила в редакцию
1. VIII. 1967 г.