

В.Ш. МЕЛИКЯН, Д.Д. ОГАНЕСЯН

**АЛГОРИТМ МИНИМИЗАЦИИ ЗАДЕРЖЕК КРИТИЧЕСКИХ
ПУТЕЙ ЦИФРОВЫХ СХЕМ**

Дан анализ эффективности существующих алгоритмов оптимизации цифровых схем при разных технологических процессах. С целью уменьшения задержек критических путей предложен метод посттопологической оптимизации цифровых схем посредством реструктуризации и перераспределения логических схем.

Ключевые слова: посттопологическая оптимизация, критический путь, минимизация задержек межсоединений.

В CMOS субмикронной технологии паразитные эффекты межсоединений стали доминирующими факторами работы схемы. Минимальный размер транзистора уменьшился до 0,1 мкм и ожидается, что скоро достигнет 0,07 мкм, что резко повышает степень интеграции. С другой стороны, становится очевидным, что рабочие характеристики схем будут в большей степени зависеть от характеристик межсоединений. Другой особенностью межсоединений является рост паразитной емкости между двумя соседними линиями. Эта емкость стала главной компонентой в суммарной паразитной емкости линии, что обусловлено увеличением соотношения геометрических размеров линий и уменьшением расстояния между ними. Из рис. 1а видно, что отношение емкости связи к суммарной емкости межсоединения (Металл 4) при минимальном расстоянии между соседними линиями увеличивается от 40 до 70% [1]. При двойном расстоянии линий это отношение увеличивается от 15 до 40%. Следовательно, при проектировании субмикронных схем весьма важен учет расстояний межсоединений.

Из вышеизложенного следует, что учет параметров межсоединений играет решающую роль в современном процессе проектирования интегральных схем (ИС). Другим весьма важным доказательством является зависимость задержек логического элемента и межсоединения от технологического процесса [1] (рис. 1б). Таким образом, очевидно, что оптимизация схем с учетом параметров межсоединений является актуальной задачей. В настоящее время уже существует много алгоритмов посттопологической оптимизации цифровых схем, которые или пренебрегают задержками межсоединений, или же используют слишком грубые значения временных параметров, в результате чего погрешность в расчетах временных моментов переключений схемы может достичь 15% [1]. Поэтому для решения вышеуказанных проблем необходимо применять более сложные алгоритмы.

Одним из путей решения отмеченной проблемы является минимизация длины межсоединений. Размещение элементов оказывает большое влияние на длину межсоединений. Следовательно, применение алгоритмов размещения элементов с учетом заранее заданных ограничений на временные характеристики схемы может в некоторой степени улучшить ситуацию. Минимизация длины межсоединения может быть осуществлена созданием дерева Стейнера для каждого межсоединения [2]. Одновременно необходимо учитывать и сопротивления межсоединений. В любом случае учет реальных топологических параметров необходим. Первым шагом оптимизации топологии межсоединения является минимизация или контролирование длины критических путей. Критическим является тот путь, который имеет

наибольшую задержку. При дальнейшей оптимизации подразумевается использование более точных моделей задержек, типа модели Элмора [3].

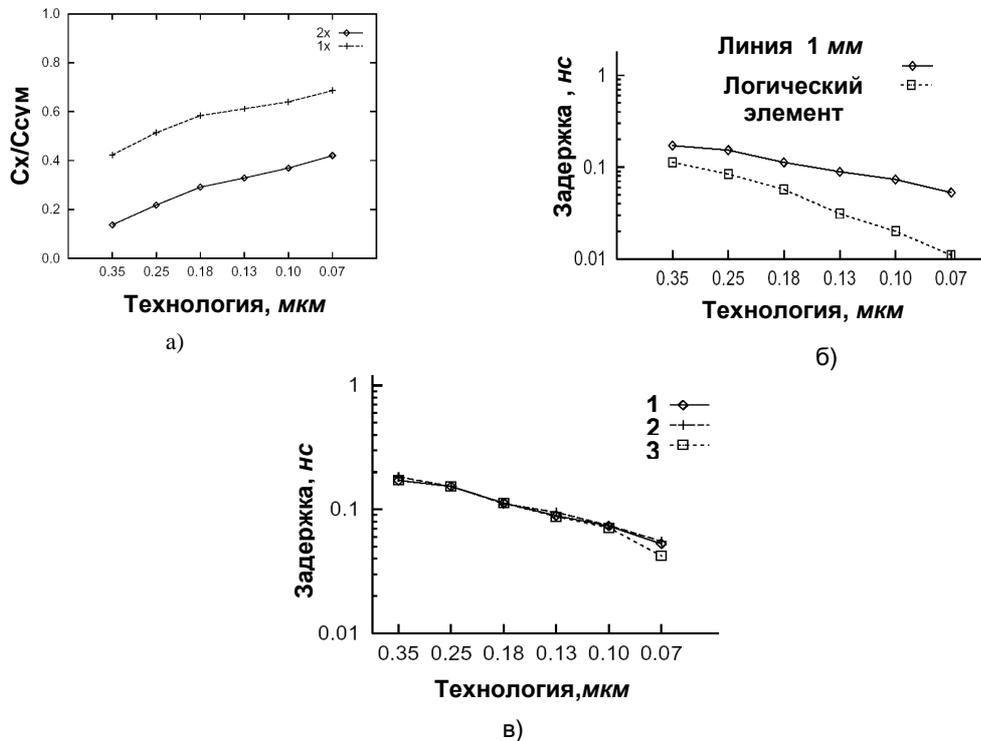


Рис. 1. Отношение емкости связи к суммарной емкости межсоединения (Металл 4) с минимальной (1x) и двойной (2x) шириной и расстоянием (а); зависимости задержек логического элемента и межсоединений от технологического процесса (б) при разных оптимизационных алгоритмах (в)

Другим путем решения проблемы минимизации задержки является учет размеров логических элементов. Если известна емкостная нагрузка межсоединения, то размеры ведущего логического элемента могут быть оптимизированы с целью минимизации задержки. Для большой нагрузки используется каскад элементов. Задача выбора размеров элементов состоит в определении каскада элементов и размеров каждого элемента. Если использовать простую RC-модель, а также игнорировать выходной емкостью ведущего элемента и межсоединений элементов каскада, то можно показать, что при емкостной нагрузке, равной C_L , и числе каскадов N отношение двух последующих элементов каскада будет равно $(C_L / C_0)^{1/N}$. Это отношение не должно меняться, чтобы обеспечить минимальную задержку. Если N не является постоянной величиной, то оптимальное отношение разрядов равно $f=e$, а число каскадов - $N = \ln(C_L / C_g)$.

В общем случае может быть использован также метод, основанный на оценке и задании размеров транзисторов с целью определения оптимальных размеров каждого транзистора для оптимизации характеристик общей схемы.

Другим эффективным способом уменьшения задержек межсоединений являются метод введения буферов [4], а также метод оценки и задания размеров соединений. Известно

несколько подходов последнего метода, но все они оптимизируют размеры отдельного соединения, в результате чего емкость связи между соседними линиями пренебрегается. В [5] предложен алгоритм, в котором этот недостаток исправлен.

Более эффективным подходом считается объединенная оптимизация элементов и межсоединений. Такой метод представлен в работе [6], в котором целевая функция состоит в минимизации суммарной задержки.

Для каждого технологического процесса применяются три оптимизационных алгоритма с целью минимизации задержки межсоединения линии длиной 1 мм. Эти алгоритмы используют следующие подходы:

1. Задание размеров ведущего элемента (драйвера).
2. Оптимальное введение буферов и задание их размеров.
3. Оптимальное введение буферов, задание размеров элементов и соединений.

Задержки оптимизированных структур межсоединений по этим алгоритмам при разных технологических процессах представлены на рис.1в. Выбор вышеуказанных оптимизационных алгоритмов обусловлен их применением в программных системах для физического проектирования и логического синтеза. Сравнительные результаты рассматриваемых оптимизационных методов показывают, что влияние размеров буферов и соединений незначительно после правильного задания размеров драйвера, даже для технологий ниже 0,1 мкм. Судя по графику (рис. 1в), лучший результат показывает метод 3, который представляет собой совокупность нескольких алгоритмов оптимизации. Но даже при этом методе улучшение задержки межсоединения незначительно. Поэтому становится ясно, что нужны новые подходы для решения проблем с задержками межсоединений.

Предлагаемый алгоритм направлен, в частности, на минимизацию задержек критических путей схемы с минимальным увеличением потребляемой мощности по сравнению с другими методами.

Заданы схема с уже готовой топологией, технологическая библиотека элементов. Предполагается, что все входные сигналы поступают в нулевой момент времени и все выходные сигналы должны быть установлены через время t_c . Цель алгоритма состоит в минимизации t_c посредством реструктуризации и перераспределения логических элементов схемы. Блок-схема предлагаемого метода приведена на рис.2а. После временного анализа схема разделяется на отдельные группы элементов (суперэлементов) в соответствии с критическим путем. Для каждого суперэлемента создается множество решений перераспределения элементов таким образом, чтобы работа схемы при этом не ухудшалась. Во время создания множеств используется алгоритм размещения элементов, параллельно которому рассчитываются задержки и паразитные параметры межсоединений. В течение работы этого алгоритма производится начальное распределение элементов, которое совершенствуется в последующих этапах. На этом этапе никаких топологических изменений не происходит. Лучшее решение для размещения элементов и физического исполнения схемы получается в ходе решения оптимизационной задачи. После этого топология схемы меняется в соответствии с принятым решением. На всей протяженности оптимизационных этапов рассчитываются и применяются ограничения на перераспределения элементов и временных параметров. Алгоритм делает итерации до тех пор, пока поставленная цель будет достигнута или дальнейшее улучшение станет невозможным. Первым шагом алгоритма является временной анализ с целью выявления критических путей. Почти любая программа логического синтеза имеет такую возможность.

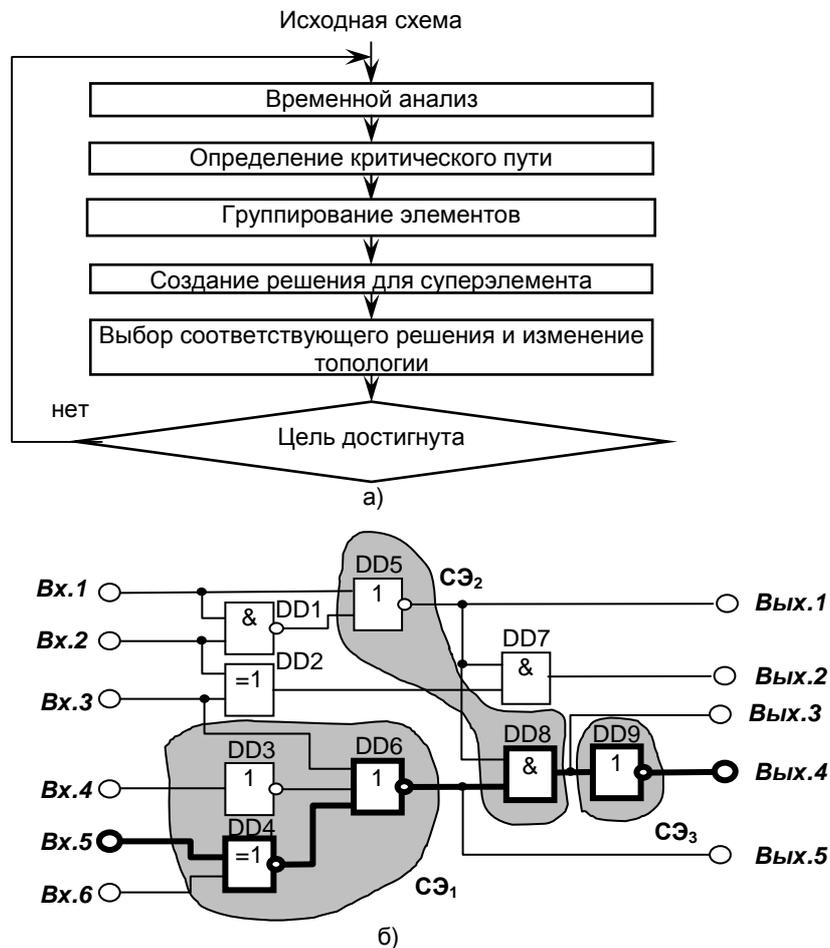


Рис.2. Блок-схема предлагаемого оптимизационного метода (а) и формирование суперэлементов (б)

Пример схемы приведен на рис. 2б. Обозначенный путь Вх5 – Вых4 является критическим. Затем схема разделяется на отдельные группы элементов – суперэлементов. Суперэлемент (СЭ) является целью для дальнейшей оптимизации. При разделении учитываются следующие допущения:

- критический элемент (КЭ) - элемент на критическом пути;
- СЭ - группа логически соединенных элементов по крайней мере с одним КЭ;
- элемент G_i является входным элементом L-уровня для G_j , если есть кратчайший путь от G_i до G_j , содержащий максимум L ветвей;
- корнем СЭ может быть только элемент, являющийся критическим и имеющий выходные ветви вне СЭ;
- среди всех выходов СЭ только один выходной порт является критическим;
- с каждого входа СЭ есть путь к критическому выходному порту.

Сначала определяются исходные СЭ, распределяемые каждому КЭ как коренной элемент. Это гарантирует, что КЭ не будет включен в другие СЭ. Затем расширяются СЭ, включая входные элементы L-уровня, если этот элемент не включен в другой СЭ. Если возникает вопрос

о присвоении данного элемента, то его получает более критический СЭ. После этого СЭ с одним выходом объединяются с другими СЭ, в результате чего размеры СЭ возрастают, не усложняя алгоритм. Поскольку критический элемент DD4 имеет только один выход, то СЭ, для которого этот элемент является корневым, сливается в СЭ с корневым элементом DD6 (рис.2). В результате образуется СЭ с двумя КЭ.

Следующим этапом алгоритма является генерирование решений для СЭ. Каждое решение содержит три параметра: R , C и τ , где C - входная емкость для входного порта, который находится на критическом пути; R - мощность корневого элемента; τ - задержка СЭ, которая включает как задержки логических элементов, так и межсоединений. Параметры R и C легко определяются из технологической библиотеки, а τ получается по ходу определения параметров СЭ соответствующими программными средствами, используя RC- или RLC- модели. Несмотря на то, что параметр площади (или потребляемой мощности) не включен в список, рассматривается соотношение выгоды и потерь площадь/задержка. Из генерированных решений исключаются все второстепенные решения. Решение $P_1(R_1, C_1, \tau_1)$ считается второстепенным по сравнению с решением $P_2(R_2, C_2, \tau_2)$, если $R_1 \geq R_2$, $C_1 \geq C_2$, $\tau_1 \geq \tau_2$. Для этого используется алгоритм, основанный на динамическом программировании. Производится перераспределение элементов и размещение с минимальным числом пересечений, как и в [7]. Нет необходимости рассчитывать параметры (R, C, τ) для каждого внешнего узла, т.к. R касается только критического выхода, а C - только критического входа. Кроме того, с целью учета соотношения выгоды и потерь площадь/задержка внесен параметр A , показывающий площадь для комбинаций, содержащих некритические узлы. В результате формируются три типа характеристик для динамического программирования: $\langle R, C, \tau \rangle$ - для корневого узла, $\langle C, \tau \rangle$ - для любого критического узла, $\langle A, \tau \rangle$ - для остальных узлов. В течение перераспределения СЭ задержка межсоединения разветвления некритического порта P может меняться. Если эта задержка станет слишком большой, данный путь может стать критическим. Чтобы предотвратить создание нового критического пути, применяются соответствующие ограничения. Выбор лучшего решения для каждого СЭ может привести к ухудшению временных параметров схемы из-за сложных зависимостей между СЭ. По этой причине лучшие решения для перераспределения и топологического исполнения всех СЭ выбираются одновременно. Для этого применяется обобщенное геометрическое программирование, которое представляет собой ряд нелинейных оптимизационных задач, имеющих целевые функции и ограничения в виде полиномов. Одним из требований этого метода является непрерывность переменных. Поэтому первым шагом необходимо превратить решения для СЭ в непрерывные, поскольку на этом этапе они дискретны. Для этого применяется интерполяция Лагранжа. Интерполированную функцию можно представить как

$$\tau = F(C, R). \quad (1)$$

Постановка оптимизационной задачи следующая:

минимизация t_c

$$a_j \geq a_i + d_{ij} \quad \forall \text{СЭ},$$

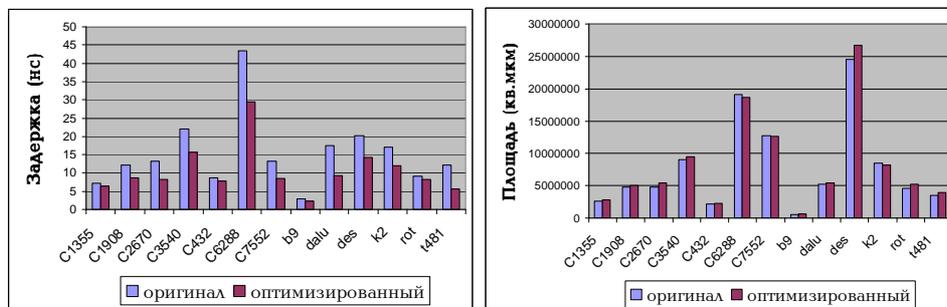
$$a_j \leq t_c,$$

$$a_j \geq 0,$$

$$|x_i - x_i'| \leq \Delta_x \quad \forall \text{СЭ},$$

$$|y_i - y_i'| \leq \Delta_y.$$

Здесь a_i, a_j – времена поступлений сигналов на соответствующие узлы; d_{ij} – задержка между этими узлами; x, y – позиционные координаты узла; Δ_x, Δ_y – ограничения перераспределения. Каждый СЭ имеет четыре переменные – x, y, R и C . Таким образом, число



переменных будет $4XN$, где N – число СЭ в схеме.

Рис. 3. Сравнительные диаграммы задержек и площадей до и после оптимизации

На последнем этапе по полученным результатам, используя соответствующее программное обеспечение, меняется топология схемы.

Для оценки эффективности разработанного алгоритма были проведены эксперименты на серии тестовых цифровых схем ISCAS 85. Схемы, использованные для оценки алгоритма, были реализованы с помощью цифровой библиотеки с минимальным размером затвора КМДП транзистора 130 нм. В состав цифровой библиотеки входили как простые логические вентили (двух-, трех- и четырехходовые И, И-НЕ, ИЛИ, ИЛИ-НЕ), так и двухтактные D-триггеры. Рабочая частота использованной библиотеки составляла 1 ГГц. В начале эксперимента были произведены распределение элементов схем и трассировка с использованием алгоритмов с временными параметрами. Для этого были использованы временные модели элементов библиотеки в формате TLF, ALF, LIB, поведенческие модели в формате Verilog и физические модели в формате LEF. Для посттопологического моделирования были использованы RLC-модели паразитных параметров. Затем определялись критические пути. Полученные результаты оптимизированы по предложенному алгоритму. Опыты проведены на системе Sun Ultra-Sparc с использованием соответствующих программ проектирования цифровых схем. Экспериментальные результаты показали, что при использовании предлагаемого алгоритма за счет потерь площади всего 5% (у аналогичных алгоритмов минимум 7...9%) наблюдается улучшение задержки схемы в 29% (у аналогичных алгоритмов максимум 15...18%) (рис.3). Этот

показатель почти в два раза превышает аналогичные результаты существующих алгоритмов. В дальнейшем этот алгоритм может быть интегрирован с другими алгоритмами оптимизации, поскольку, как было показано, при объединении отдельных методов выигрыш получается больше, чем сумма выигрышей от применения отдельных методов.

СПИСОК ЛИТЕРАТУРЫ

1. **Dudzinski R.** RTL Floorplanning Set to Drive Synthesis // EE Times. –1995. - № 27. –P. 52.
2. **Cong J., He L., Koh C. and Madden P. H.** Performance optimization of VLSI interconnect layout// Integration, the VLSI Journal. -1996. –V. 21. –P. 1-94.
3. **Elmore W. C.** The Transient Response of Damped Linear Networks // Journal of Applied Physics. – January, 1948.-Vol. 19. -P. 55 - 63.
4. **Ginneken L. P. P. P.** Buffer placement in distributed RC-tree networks for minimal Elmore delay // In Proc. IEEE Int. Symp. on Circuits and Systems. -1990. –P. 865–868.
5. **Cong J., He L., Koh C. and Pan Z.** Global interconnect sizing and spacing with consideration of coupling capacitance // In Proc. Int. Conf. on Computer Aided Design. -1997. – P. 570-573.
6. **Cong J., Koh C. and Leung K.** Simultaneous buffer and wire sizing for performance and power optimization // In Proc. Int. Symp. on Low Power Electronics and Design. - Aug. 1996. – P. 271–276.
7. **Lou J., Salek A. H. and Pedram M.** An Exact Solution to Simultaneous Technology Mapping and Linear Placement Problem // In Proc. of International Conference on Computer Aided Design. - 1997. – P 671-675.

ГИУА. Материал поступил в редакцию 10.01.2003.

Վ.Շ. ՄԵԼԻԿՅԱՆ, Դ.Դ. ՀՈՎՀԱՆՆԻՍՅԱՆ
ԹՎԱՅԻՆ ՄԻՆԵՄԱՆԵՐԻ ՈՐՈՇԻՉ ՃԱՆԱՊԱՐՀՆԵՐԻ ԺԱՄԱՆԱԿԱՅԻՆ
ՀԱՊԱՂՈՒՄՆԵՐԻ ՆՎԱԶԱՐԿՄԱՆ ԱԼԳՈՐԻԹՄ

Դիտարկված են թվային ինտեգրալ սխեմաների նախագծման ընթացքում ծագող միջմիացումների հետ կապված խնդիրները տարբեր տեխնոլոգիական գործընթացների դեպքում: Բերված է լավարկման տարբեր ալգորիթմերի համեմատական վերլուծությունը: Առաջարկված է թվային ինտեգրալ սխեմաների հապաղումների նվազարկման նոր ալգորիթմ՝ հիմնված տարբերի վերադասավորման և տեղաբաշխման վրա:

V. SH. MELIKYAN, D. D. HOVHANNISYAN
DIGITAL CIRCUIT MINIMIZATION ALGORITHM OF CRITICAL PATH DELAYS

The efficiency of existing algorithms for digital circuit optimization at different technologies are analyzed. To decrease the critical path delays, a method of post-topological optimization of digital circuits by means of restructuring and placement of logical circuits has been proposed.