

Prace Instytutu Technologii Elektronowej, z.3, s.1 - 8, Warszawa 1983

Adam Wyszyński

Zakład Projektowania Układów Scalonych ITE

Warszawa

AUTOMATYCZNE PROJEKTOWANIE LAYOUTU UKŁADU MOS LSI
METODĄ DYNAMICZNĄ

(Rękopis dostarczono 12 listopada 1982)

S t r e s z c z e n i e

W komunikacie opisano metodę projektowania layoutu układu scalonego. Polega ona na rozmieszczaniu bloków podstawowych i prowadzeniu połączeń w dwóch warstwach. Główna cecha prezentowanej metody jest komponowanie layoutu według zasad programowania dynamicznego: należy dokonać wstępnego rozmieszczenia typu "blok po bloku" z jednoczesnym wytyczaniem tras połączeń, a następnie poprowadzić połączenie i ostatecznie rozmieścić bloki podstawowe.

1. Wprowadzenie

W procesie projektowania layoutu układów MOS LSI szczególnie pracochłonnym i uciążliwym etapem jest rozmieszczenie zaprojektowanych uprzednio bloków podstawowych oraz dokonanie takiego ich połączenia, aby powierzchnia zajmowana przez bloki i połączenia była jak najmniejsza. Na optymalnie zaprojektowany layout wpływa odpowiednie ułożenie prostokątnych bloków w taki sposób, aby straty powierzchni wynikające z ich niedopasowania były jak najmniejsze. Ma także wpływ odpowiednia kolejność rozmieszczania bloków /zapewniająca minimalną długość połączeń związanych z każdym z nich/ oraz prowadzenie każdego połączenia po najkrótszej drodze.

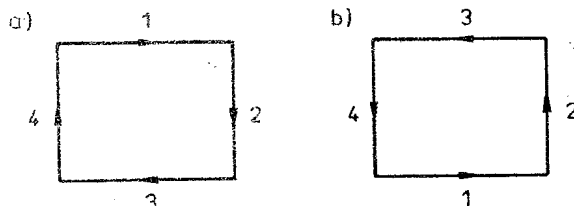
Optymalnie zaprojektowany layout, poprzez zmniejszenie zajmowanej powierzchni, może spowodować wzrost uzysku produkcyj-

nego, a także poprzez skrócenie najdłuższych ścieżek, może zwiększyć częstotliwość pracy układu. Najważniejszą cechą automatycznego rozmieszczania i prowadzenia połączeń w układach MOS LSI jest jednak wydajne skrócenie czasu powstawania projektu, który maleje z kilku do kilkunastu tygodni przy metodzie ręcznej, do kilku lub kilkunastu godzin - przy projekcie automatycznym.

2. Założenia ogólne

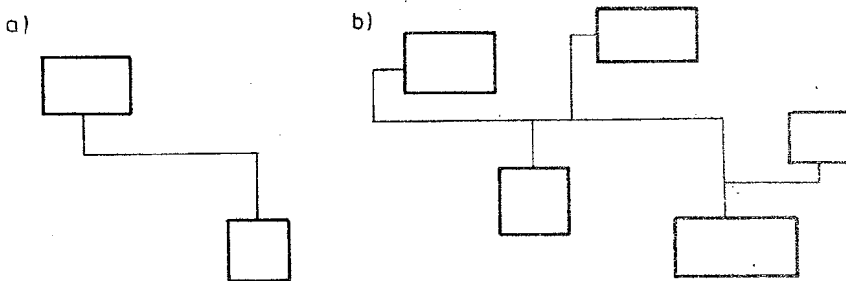
Założenia wynikają z konkretnych potrzeb projektowych, niektóre z nich można dość łatwo zmienić lub uzupełnić /w zależności od rodzaju projektowanego układu/. Podstawowe założenia są następujące:

- 1/ mapa układu ma kształt kwadratu, pola montażowe są rozmieszczone równomiernie na wszystkich jej bokach,
- 2/ sekwencja pól montażowych jest z góry zadana,
- 3/ bloki podstawowe są prostokątami; każdy z nich może mieć różne wymiary,
- 4/ położenie wyprowadzeń z bloku podstawowego jest z góry zadane,
- 5/ blok podstawowy może znaleźć się na layoucie w dwóch wersjach: wersji podstawowej i wersji odbicia lustrzanego /rys. 1/.



Rys. 1. Dwie wersje bloku podstawowego: a) wersja podstawowa, b) wersja odbicia lustrzanego /strzałkami zaznaczono kolejność wyprowadzeń na boku/

- 6/ dopuszcza się obroty obu wersji bloku o kąt $\alpha = k \cdot \pi/2$, gdzie $k = 0, 1, 2, 3$, /daje to w sumie 8 różnych kombinacji współrzędnych wyprowadzeń/,
- 7/ układ posiada co najmniej dwie warstwy połączeń,
- 8/ połączenia mogą być zarówno typu zwykłego - łączące dowolne dwa kontakty /rys. 2a/, jak i typu rozgałęzionego - łączące więcej niż dwa kontakty /rys. 2b/,



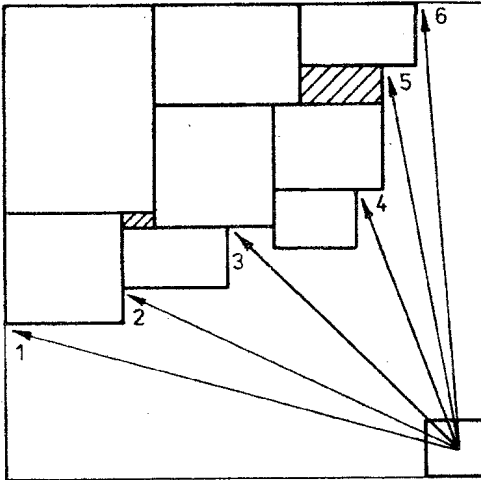
Rys. 2. Typy połączeń w układzie: a) połączenia zwykłe, b) połączenia rozgałęzione .

- 9/ wyróżnia się połączenia typu masa, zasilanie oraz inne; dopuszcza się różne szerokości połączeń, a w szczególności możliwe jest żądanie, aby niektóre z połączeń uczynić możliwie krótkimi,
- 10/ dopuszcza się istnienie wprowadzonych celowo bloków bez wyprowadzeń, a także "sztucznych" połączeń wiążących dodatkowo bloki mające znaleźć się na layoucie blisko siebie,
- 11/ ogranicza się liczbę bloków podstawowych do 50, liczbę półmontażowych do 48, liczbę połączeń do około 500, w zależności od ich typu i liczby łączonych kontaktów /ograniczenia te można podwyższyć poprzez zwiększenie obszaru zajmowanej pamięci/.

3. Opis metody

Na początku przedstawia się obszary które leżą pomiędzy bokami bloków podstawowych jako odcinki. Nazwano je kanałami

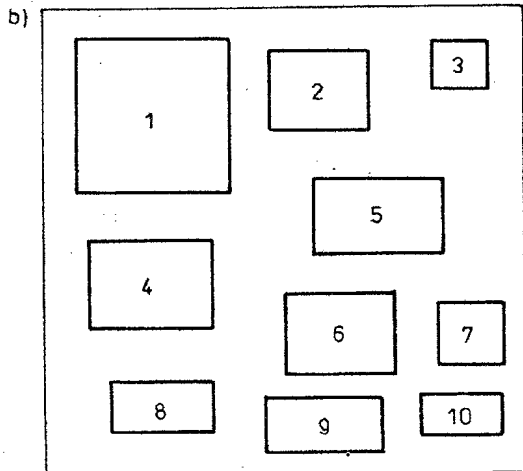
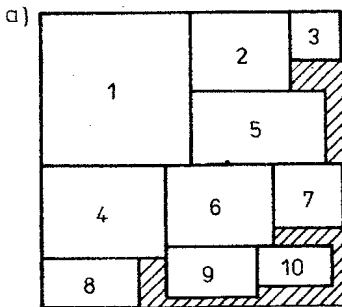
połączeń. Umieszczając kolejny blok bada się jego dopasowanie do rozmieszczonych już bloków, a także wytycza trasy połączeń do rozmieszczonych już kontaktów oraz oblicza się ich wstępną długość. Na podstawie obu czynników /czynnika geometrycznego i czynnika połączeń/ wyznacza się funkcję celu. Funkcję tę oblicza się: dla wszystkich miejsc, w których blok może być umieszczony, dla wszystkich nie rozmieszczonych bloków, dla każdej z danych ośmiu orientacji kontaktów bloku /rys. 3/. Ocenia się, że czas obliczeń dla



tego etapu jest proporcjonalny do $k \cdot n^a$, gdzie: k - średnia liczba połączeń na blok, n - liczba bloków, $1,5 \leq a \leq 2$.

Rys. 3. Wstępne rozmieszczanie typu "blok po bloku"

tego etapu jest proporcjonalny do $k \cdot n^a$, gdzie: k - średnia liczba połączeń na blok, n - liczba bloków, $1,5 \leq a \leq 2$.



Rys. 4. Dwa etapy projektowania: a) wstępne rozmieszczenie, b) końcowe rozmieszczenie po poprowadzeniu połączeń

Na rys. 3 powierzchnia zakreskowana oznacza powierzchnię straconą powstałą w wyniku niedopasowania wymiarów bloków. Odcinki, z których składają się ściany bloków, to kanały połączeń. Cyframi /1 - 6/ oznaczono miejsca, w których można umieszczać bloki.

Przy wytyczaniu tras połączeń dąży się do zapewnienia podobnego zagęszczenia kanałów przez ścieżki. Ma to duże znaczenie dla uniknięcia strat powierzchni w następnym etapie, kiedy po poprowadzeniu połączeń bloki zostaną rozmieszczone ostatecznie. Jeżeli zapewni się, że szerokości kanałów będą podobne, to zoptymalizowane wstępne rozmieszczenie nie zostanie zmienione w zasadniczy sposób, powiększą się jedynie szerokości odstępów między blokami /rys. 4a, b/.

4. Projektowanie layoutu jako zadanie programowania dynamicznego

Programowanie dynamiczne polega na przedstawieniu wieloetapowego zagadnienia decyzyjnego jako ciągu zagadnień jednoetapowych, przy czym każda decyzja cząstkowa powinna być zgodna z poprzednimi /w sensie realizowania pewnej polityki optymalizacji/.

Postać podstawowa zależności rekurencyjnej dla naszego zagadnienia jest następująca:

$$Ss_{i+1} = \min_k \left(w_i^k + Ss_i \right), \quad (1)$$

gdzie: Ss_{i+1} - powierzchnia strat po przyłączeniu i-tego bloku, Ss_i - powierzchnia strat przed przyłączeniem i-tego bloku, w_i^k - powierzchnia strat wynikających z niedopasowania wymiarów bloku i wydłużenia połączeń na skutek niewłaściwej kolejności umieszczania bloku.

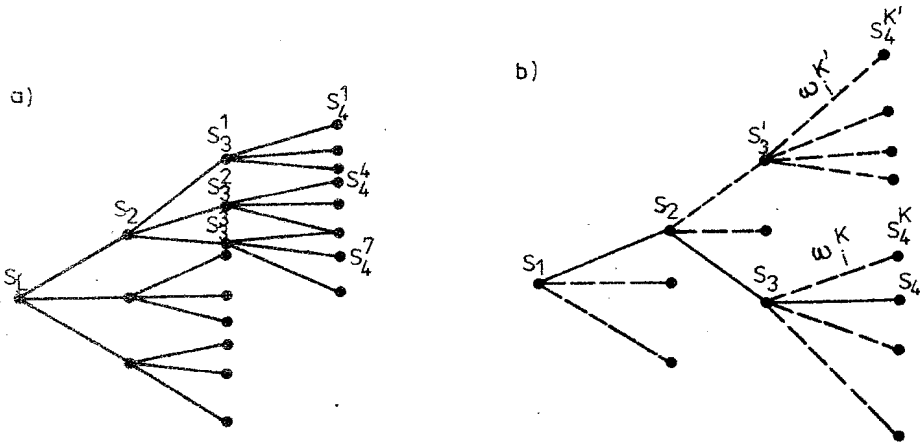
Można wykazać, że w_i^k nie zależy bezpośrednio od drogi przejścia układu od stanu początkowego do stanu i-tego, lecz jest bezpośrednio związane ze stanem i+1 układu, oraz z przyłączanym blokiem. Oznacza to, że warunek Bellmana jest spełniony i zagadnienie staje się rozwiązywalnym zadaniem programowania dynamicznego.

Ponadto można wykazać, że pełna zależność rekurencyjna (1) może być zastąpiona zależnością uproszczoną

$$Ss_{i+1} = \min_k (w_i^k) + Ss_i. \quad (2)$$

Oznacza to poważne uproszczenie zagadnienia, które w pełnej postaci charakteryzuje się potężnym wzrostem liczby rozważanych przypadków.

Poprawność uproszczenia jest uzasadniona takim doborem funkcji sterującej w_i^k , aby prawdopodobieństwo tego, że gdy $S'_1 > S_1$ to $S'_1 + w_1^{k'} < S_1 + w_1^k$, było niewielkie /rys. 5/.



Rys. 5. Postacie funkcji stanu: a) pełna postać funkcji stanu dla zadania programowania dynamicznego /np. możliwe jest $S_4 < S_4$ mimo, że $S_3 > S_3$ /, b) uproszczona postać funkcji stanu /prawdopodobieństwo, że dla któregoś z przypadków $S_4 < S_4$ jest niewielkie/

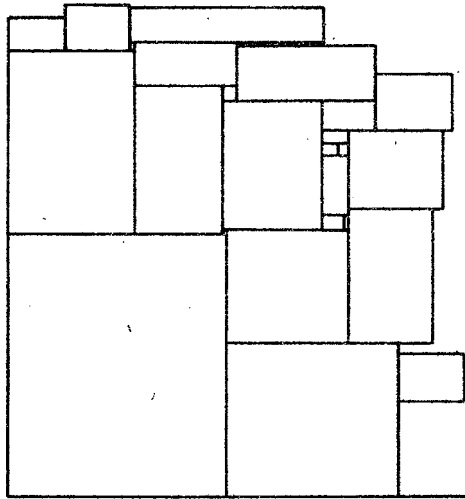
Ostatecznie po n-krokach, czyli po umieszczeniu wszystkich bloków, korzystając z (2) mamy

$$Ss_n = \sum_{i=1}^n \left(\min_k (w_i^k) \right), \quad (3)$$

czyli Ss_n jest sumą minimalnych strat powierzchni w układzie.

5. Dotychczasowe wyniki eksperymentalne

Na podstawie prezentowanej metody opracowano system projektujący layout. System uruchamiany jest na EMC ODRA 1305 pod systemem operacyjnym GEORGE 3. Istnieje możliwość oceny wyników pośrednich poprzez wykonanie rysunków na urządzeniu rysującym.



Rys. 6. Wstępne rozmieszczenie 19 bloków bez uwzględnienia sieci połączeń. Czas obliczeń około 1 minuty /rysunek jest symetryczny w stosunku do poprzednich/

Eksperymenty związane z częścią geometryczną rozmieszczenia potwierdziły przydatność funkcji celu do tego zadania. Powyższy rysunek /rys. 6/ został wygenerowany przez maszynę. Funkcja celu opisuje w tym przypadku jedynie zależności geometryczne.

6. Podsumowanie

Wydaje się, że powyższa metoda znalazłaby zastosowanie w projektowaniu layoutów, układów wykonywanych w dowolnej technologii dopuszczającej wielowarstwowe połączenia. Ze wzglę-

du na rozdzielenie etapu rozmieszczania od etapu prowadzenia połączeń, do tego ostatniego można zastosować różne techniki. Posługiwanie się funkcją celu w danym problemie stwarza możliwości optymalizacji innych wielkości /np. czasów opóźnień sygnałów/.

L i t e r a t u r a

1. W i l l i a m s J.D.: STIKS-A Graphical Compiler for High-Level LSI Design. AFIPS Conference Proceedings June 1978 vol. 47 p. 289-295.
2. D u n l o p A.F.: SLIP: Symbolic Layout of Integrated Circuits with Compaction. Computer-Aided Design 1978 vol.10 nr 6 p. 387-391.
3. H s u e h N.Y, P e d e r s o n D.O.: Computer-Aided Layout of LSI Circuit Building-Blocks. IEEE International Symposium on Circuits and Systems 1979 p. 474-477.
4. A b r a i t i s L., B l o n s k i s I., K u ź m i o z W.: The Method of Computer-Aided Layout Design for Integrated Circuits. Design Automation and Fault - Tolerant Computing Winter 1979 vol. 3,4 p. 191-209.
5. K u b i a k R.: /konsultacje/.

A METHOD OF MOS LSI LAYOUT AUTOMATIC DESIGN

/ Received November 12, 1982/

Abstract

The paper describes a method of an integrated circuit layout design. It consists in placing building blocks and leading connections on two layers. The main feature of presented method is layout composition due to the dynamic programming rules: at first initial placement of "block after block" type together with simultaneous connection tracing is performed, than connections are led and building blocks are finally placed.

1. Introduction

During MOS LSI layout design process the most laborious step is to place already designed building blocks and to connect them in the way that the *area* occupied by blocks and connections is the smallest. The optimum *designed* layout results from proper arrangement of rectangular blocks, so that the *area* loss derived from their unfitting is minimized. The placement sequence providing minimum connection length led from each block is of big importance too.

The optimum designed layout having smaler *area* can cause the production yield increase and moreover, due to shortening the longest paths, can increase the working frequency of the circuit.

Yet, the most important feature of automatic placement and routing in MOS LSI circuits is the distinctive shortening of ~~the~~ project time, which decreases from several weeks in hand design to several hours in automatic design.

2. Principles

They derived from specific design requirements. Some of them can be easily replaced or supplemented /according to the type of designed circuit/.

The basic principles are as follows:

- 1/ circuit map is square, bonding pads are regularly placed on all its sides.
- 2/ bonding pad sequence is given in advance.
- 3/ building blocks are rectangular, each of them can have arbitrary dimensions.
- 4/ the positions of contacts on building blocks are fixed.
- 5/ there are two basic version of a building block available on the layout: basic and mirrored one.

Fig. 1. Two versions of a building block: a/ basic version,
b/ mirrored version /the sequence of contacts on the side is marked with a flash/.

6/ rotations of two block versions at an angle of $\alpha = k \cdot \pi / 2$, where $k=0,1,2,3$ are allowed./it gives 8 different combinations of contact coordinates/.

7/ the circuit has got at least two connection layers.

8/ connections can be simple -that connects two given contacts /fig. 2a/, as well as branched -that connects more than two contacts /fig. 2b/.

Fig. 2. Connection types in the circuit: a/ single connection
b/ branched connection.

9/ ground, supply and other types of connections are distinguished, different connection widths are allowed, in particular making a claim is possible to have certain connections the shortest.

10/ blocks without contacts, introduced on purpose are allowed as well as "artificial" connections additionally binding the blocks which are too close one another.

11/ the number of building blocks is limited to 50, the number of bonding pads to 48, the number of connections approximately to 500, according to their type and the number of connected contacts. /this limits can be increased by the memory growth/.

3. The description of the method

At the beginning the areas which are laid between building blocks sides are represented by sectors. They are called connection channels. Placing the next block, fitting to already placed blocks is analysed and connection tracks to already placed contacts are traced as well as their approximate length is calculated. On the base of both factors /geometry and connection length/ the purpose function is assigned. This function is calculated for all places a block can be placed, for all not placed blocks, for given one /of possible eighth/ orientation of contacts. /fig. 3/.

It is estimated that calculation time for this stage is proportional to $k \cdot n^a$ where: k -the average number of block connections, n -the number of blocks, $1.5 \leq a \leq 2.0$.

Fig. 4. Two design stages: a/ initial placement b/ final placement after connections are led.

Fig. 3 shows shadowed loss surface created as the result of block dimension unfitting. The sectors, which block sides are consisted of, form connection channels. By digts /1-6/ placement sites are marked.

While tracing connections an attempt is made to assure the

similar congestion in channels. This is of great importance for avoiding *area* loss in the next stage, when after leading connections, blocks are finally placed. If similar channel widths are assured, then the initial optimum placement is not strongly changed, separation between blocks increase only /fig. 4 a, b/.

4. Layout designing as the dynamic programming problem

The dynamic programming consists in describing a multistage decision problem as a sequence of one-stage problems, where each partial decision should be consistent with previous /in sense of an optimization policy realization/.

Basic recursive form for our problem is as follow:

$$S_{S_{i+1}} = \min_k (\omega_i^k + S_{S_i}) \quad (1)$$

where: $S_{S_{i+1}}$ - loss surface after connecting of i -th block, S_{S_i} - loss surface before connecting of i -th block, ω_i^k - *area loss* resulting from unfitting block dimensions and increasing connection length because of improper placing sequence.

It can be showed that ω_i^k is not direct dependent on the way the system is passing from initial state to i -th state, but it is directly attached to $i+1$ -th state of system and to the connected block. It means that Bellman's condition is satisfied and the problem is the solvable dynamic programming problem.

Moreover, it can be showed that the full recursive form (1) can be substituted by a simplified one.

$$S_{S_{i+1}} = \min_k (\omega_i^k) + S_{S_i} \quad (2)$$

It means an important simplification of the problem, which in the full form is characterized by an exponential growth of discussed cases.

Correctness of this simplification is justified by such a choice of the control function ω_i^k , for which probability that if $S_i' > S_i$ then $S_i' + \omega_i^{k'} < S_i + \omega_i^k$ is small /fig. 5/.

Fig. 5. The forms of the purpose function a/ the full form for the dynamic programming problem /e.g. it is possible that $S_4^7 < S_4^4$ in spite of $S_3^3 > S_3^2$ /, b/ simplified form /probability that for one of cases $S_4^1 < S_4$ is small/.

Finally, after n steps or having placed all blocks, using (2) we obtain:

$$S_{S_n} = \sum_{i=1}^n \left(\min_k (\omega_i^k) \right) \quad (3)$$

thus S_{S_n} is the minimum sum of *area loss* in the circuit.

5. Hitherto experimental results

On the basis of presented method a layout design system has been worked out. The system is developed on ICL 1900 computer under GEORGE-3 control. There is possibility to estimate intermediate results by means of plotter drawings.

Fig. 6. Initial placement of 19 blocks without taking into consideration connection net. The CPU time is approx. 1 min /this drawing is symmetrical to the previous/.

Experiments related to geometrical aspect of placing confirmed usefulness of the purpose function in this task. The above drawing /fig. 6/ was computer generated. In this case the purpose function describes geometrical relation only.

6. Summary

It seems that the above method could be applied to layout design of circuits made in any technology admitting multilayer connections. As placement and routing are treated separately the routing can be performed with use of various techniques. Employing the purpose function in a given design creates possibility to optimize other quantities /e.g. delay times of signals/.

Biography

The biography and all drawings the reader can find in the Polish text.