論文 85-22-1-11

A Channel Routing System using CMOS Standard Cell Library

(CMOS 표준 Cell Library를 이용하는 수평 트랙 배선 시스템)

鄭 泰 聖*, 慶 宗 旻*

(Tae Sung Jung and Chong Min Kyung)

要 約

이 논문에서는 standard cell의 layout을 위한 doglegging을 하지 않는 channel 배선 시스템에 대하여 서술할 것이다. 이 시스템은 주어진 net list specification을 만족시키기 위하여, 각각 standard cell 의 직선 배열 결합인 두 row 사이의 수평 track에서 이층의 최종 배선 패턴을 만들어 준다. 이 논문에서 사용한 CMOS cell library는 9개의 기본 cell을 가지고 있으며, Mead-Conway 방식에서의 λ=2 micron 을 사용하여 CIF(Caltech Intermediate From) 형태로 표현되었다. Component library에는 각 cell 내의 pin들의 이름, 위치 및 layer type 등의 입출력 port 특성이 저장되어서, CROUT라는 channel routing program에서 입력 자료로 사용된다. 또 다른 program NETPLOT은 routing 결과를 개략적으로 도시하여 주며, NETCIF에서는 최종의 자세한 layout을 CIF file로 만들어 주고 있다. 기본 cell을 이용하여 set/reset이 있는 dynamic master-slave형 D flip-flop에 대한 channel routing의 경우 VAX 11/780 에서 4초의 CPU 시간이 소요되었다.

Abstract

In this paper, we present a non-doglegging channel routing system for IC layout using standard cells. This system produces a final two-layer wiring pattern in the horizontal track between two rows, each of which is a linear placement of standard cells of identical heights, satisfying the given net list specification. The layout of CMOS cell library including nine primitive cells used in this paper is represented in CIF (Caltech Intermediate Form) using λ(Lambda) of 2 microns in Mead-Conway layout representation scheme. The cell dimension and I/O characteristics such as name, position and layer type of the pins are stored in Component Library to be used in the channel routing program, CROUT. A subprogram, NETPLOT, was used to report a schematic layout result, and another subprogram, NETCIF, was used to yield a full-fledged final layout representation in CIF, A test run for realizing a dynamic master-slave D flip-flop with set/reset using primitive cells was shown to take 4 CPU seconds on VAX 11/780.

*正會員, 韓國科學技術院 電氣및 電子工學科

(Dept. of Elec. Eng., KAIST)

接受日字 : 1984年 9月 3日

I. Introduction

The recent demand for a fast delivery of complex integrated circuits satisfying specific

system requirement has resulted in the development of various custom design methodologies such as programmable logic array, master slice approach and standard cell approach. These methods have their advantages and disadvantages relative to the others in terms of turn-around time and packing density. Standard cell approach is now being adopted by most of the major custom IC suppliers due to its high packing density and reasonable turn around time. But, it is obvious that this approach is not viable without extensive cell library and efficient placement and routing tools. The development of standard cell routing system has been made by many researchers.[1,4] In this paper, we report an on-going activity in KAIST on the development of the standard cell routing system.

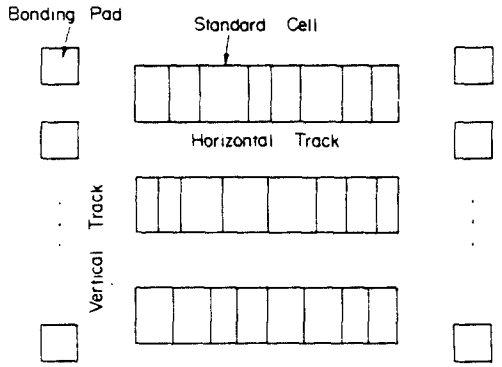


Fig. 1. Basic structure of standard cell custom chip.

In the basic structure of standard cell custom chip shown in Fig. 1, we, first of all, confined our interest in the channel routing in the region denoted as 'horizontal track,' which constitutes the main content of this paper. The cells of varying width but same height are placed in multiple horizontal rows, where the VDD and ground bus connections are usually made automatically by abutting cells side by side. Multiple row routing problem can be reduced to a set of two row routing problems, and the left(right) ports pulled out to left (right) vertical track after the two row routing are interconnected with the bonding pads according to the similar algorithm of two row routing scheme with 90 degree rotation. Generally, the horizontal channel routing problem is preceded by the network partitioning and placement which affect the ultimately achievable routing result. However, since these problems are worthy of another major effort by themselves, we will concern ourselves only with routing procedure assuming that the placement was achieved by some means. A brief explanation of the overall routing system including generation of final artwork will be given in section II. An N-well CMOS cell Library including 9 primitive cells using λ=2 micron in Mead-Conway's layout scheme[2] is explained in section III. Section IV covers the basic concept and algorithm for channel routing, which was based on the algorithm reported by Yoshimura [4]. Finally, two run examples for dynamic D-type flip-flop and BCD-to-Decimal decoder were shown in section V.

II. Standard Cell Routing System

Fig. 2 shows the overall block diagram for standard cell routing system. CROUT is a program that takes in the netlist data and placement data, and calculates the channel wiring pattern using horizontal and vertical wire segments in two conductive layers. NETCIF converts the output of CROUT into CIF format, which is, in turn, displayed on the storage type

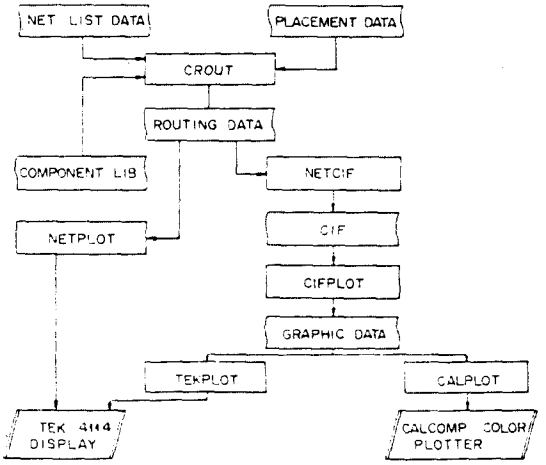
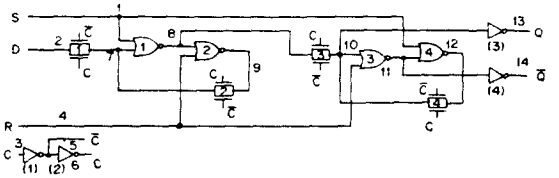


Fig. 2. Block diagram of the standard cell routing system.

display terminal or plotted on Calcomp color plotter using the graphics postprocessing routines, CIFPLOT, TEKPLOT and CALPLOT. NETPLOT is another routine which can visualize the channel wiring result on graphic terminal, but in a schematic form handling each cell as an MBB (minimum bounding box) and each wire as a line.

A D-type dynamic flip-flop with set and reset will be used as an example to illustrate the routing procedure in this system. Circuit schematic diagram for the D f/f is shown in Fig. 3(a), where the numbers denote each net, respectively (A net is defined here as a set of terminal pins of various cells tied electrically).



(a) Circuit schematic for D-FF with the device number assigned.

```
N = 14 ;
NET1  : NOR2   1  1  , NOR2   4  1  ;
NET2  : TRANS  1  2  ;
NET3  : INVER  1  1  ;
NET4  : NOR2   2  2  , NOR2   3  2  ;
NET5  : INVER  1  2  , INVER  2  1  , TRANS  1  1  , TRANS  2  3
       ,TRANS  3  3  , TRANS  4  1  ;
NET6  : INVER  2  2  , TRANS  1  3  , TRANS  2  1  , TRANS  3  1
       ,TRANS  4  3  ;
NET7  ; TRANS  1  4  , NOR2   1  2  , TRANS  2  2  ;
NET8  : NOR2   1  3  , NOR2   2  1  , TRANS  3  2  ;
NET9  : NOR2   2  3  , TRANS  2  4  ;
NET10 : TRANS  3  4  , NOR2   3  1  , TRANS  4  2  , INVER  3  1  ;
NET11 : NOR2   3  3  , NOR2   4  2  , INVER  4  1  ;
NET12 : NOR2   4  3  , TRANS  4  4  ;
NET13 : INVER  3  2  ;
NET14 : INVER  4  2  ;

LEFT (4)  : NET1,  NET2,  NET3,  NET4 ;
RIGHT(2)  : NET13, NET14 ;
END.
```

(b) Net list data for D-FF with set/ reset.

```
HIGH : INVER 1, TRANS 1,  TRANS 2,  TRANS 3,  NOR2  3, INVER 3 ;
LOW  : INVER 2, NOR2  1,  NOR2  2,  TRANS 4,  NOR2  4, INVER 4 ;
```

(c) Placement data for D-FF with Set/Reset.

Fig. 3.

The netlist data which represent the connectivity of the D f/f is shown in Fig. 3(b), where the names and numbers of nets to be pulled out to the left(right) vertical channel are shown in the bottom two lines, as well as the specification of each net with its elements. Fig. 3(c) shows the placement in two horizontal rows (the 'High' row and 'Low' row) of the standard cells to be used in realizing the D f/f. The exact location of the pins in their subscribing net, and the cell size can be obtained from the 'Component Library', whose format is shown in Fig. 4 for the case of a 2-input NOR gate.

```
COMPONENT = NOR 2 : (2 INPUT NOR)

    CIF = 440 ;
    FRAME = 60, 161 ;
    EQU = 1, 2 ;
    PIN1  : POS = 2, NAME = IN1/I ;
    PIN2  : POS = 5, NAME= IN2/I ;
    PIN3  : POS = 3, NAME = OUT/O;
```

Fig. 4. An example for component library for 2 input NOR gate.

The Component Library contains the identification number for the corresponding CIF file, the size of the MBB (minimum bounding box), the set of logically equivalent pins, and the name and position of each pin.

III. CMOS cell library

A basic set of nine primitive logic cells of identical heights constitutes our cell library for N-well CMOS process. The number of transistors contained and the width of each primitive cell are shown in Table 1. The height of the cell, which is invariant throughout the whole cell library, is 161 micron. The logic symbol for each cell is shown in Fig. 5 with its associated pin numbers.

The layout of each primitive cell is shown in Fig. 6.

It can be seen from Table 1 that the cell width is always an integer multiple of 12 micron for each cell. The pin position in each cell, which is the center x-coordinate of the

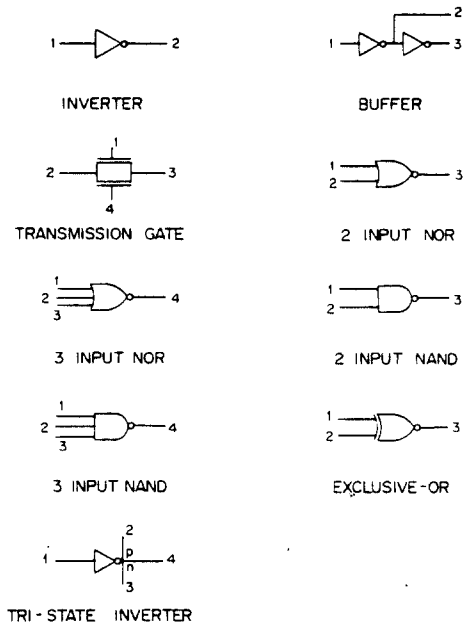


Fig. 5. Logic symbol and the associated pin number for each primitive cell.

polysilicon terminal line at the cell bottom line, has been adjusted to be some integer times of 12 micron (This integer number is what is shown as 'position' in Fig. 4).

Table 1.

Cell Name	Number of Transistors	Cell Width (Micron)
Inverter	2	48
Tri-state Inverter	4	72
Buffer	4	72
Transmission Gate	2	72
2-input NOR	4	60
3-input NOR	6	108
2-input NAND	4	60
3-input NAND	6	84
Exclusive-OR	20	216

IV. Channel Routing Algorithm

Fig. 7 shows the simplified program structure of the channel router, CROUT.

It is assumed that there is no cyclic conflict in net list. The main purpose here is to minimize the number of horizontal tracks after routing. Ultimate routing result can be achieved by repeated merging operation. The role of each procedure in CROUT is as shown below.

PREPROCESSOR: Cell information for

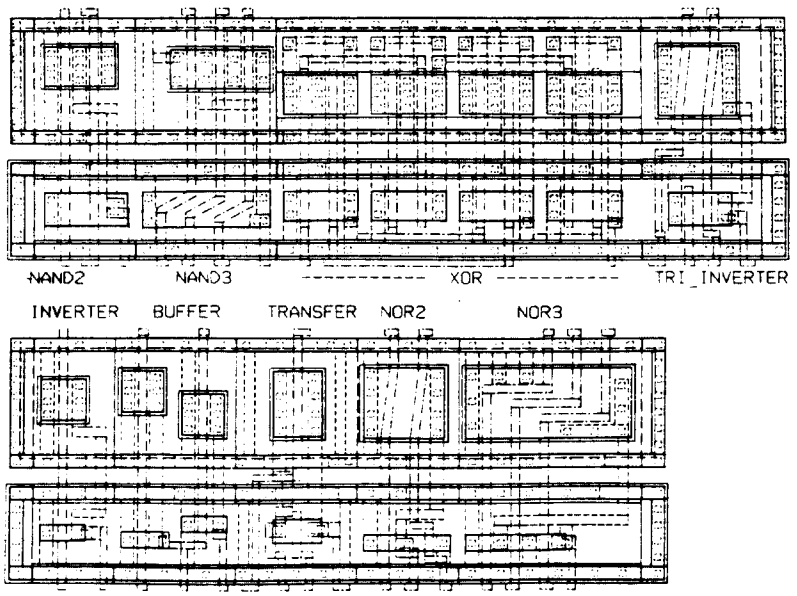


Fig. 6. Physical layout for 9 primitive cells.

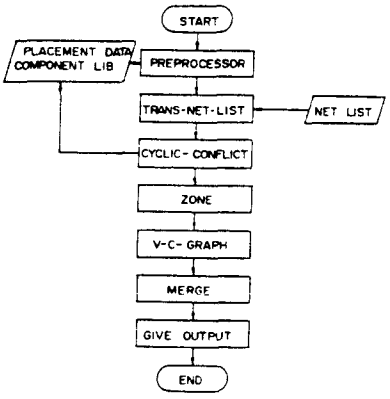


Fig. 7. Program structure of CROUT.

routing such as pin name and pin position is read from COMPONENT LIBRARY, after reading in the placement data.

TRANS-NET-LIST : Designer specifies net numbers, gate names and signal names in net list file. For the channel routing problem, it is convenient to have simplified net list composed of two rows of integer representing the net number. Procedure TRANS-NET-LIST translates textual net list to the net list representation as shown in Fig. 8 (a).

```
* NET LIST

HIGH : 0 3 0 5 6 0 2 5 0 7 5 0 7 6 0 9
       5 0 8 6 0 10 0 10 11 0 4 0 10 0 13
LOW  : 0 5 0 6 0 1 8 0 7 0 8 9 0 4 6 0
       10 5 0 12 0 1 12 0 11 0 11 0 14 0 0
```

(a) Net list data translated from Fig. 3. (b) and Fig. 3 (c).

```
* MERGED NETS AND TRACK

H TRACK NUMBER          NETS

1                :      3   7   ;
2                :      5   ;
3                :      13  6   ;
4                :      4   ;
5                :      10  2   9   ;
6                :      12  8   ;
7                :      14  11  1   ;
```

(b) Merged nets for M/S D-FF

Fig. 8.

CYCLIC-CONFLICT : Only one horizontal track is allowed for each net: as a result, nets which have a cyclic conflict can't be routed. Procedure CYCLIC-CONFLICT finds whether there exist a cyclic conflict or not.

ZONE: If we assume S(i) to be the set of nets whose horizontal segments intersect column i, the horizontal segments of any two nets in S(i) must not be placed on the same horizontal track. Moreover, it is sufficient to consider those S(i) which are not the subset of any other set. Therefore we group as many consecutive columns as possible such that every S(i) in the group is a subset or itself of some maximal S(j) in the group. We declare each such group as a zone.

V-C-GRAPH: If we assume that there is only one horizontal segment per net, then it is clear that the horizontal segment of a net connected to the upper terminal at a given column must be placed above the horizontal segment of another net connected to the lower terminal at that column. This relation is called vertical constraint and can be represented by a directed graph which is called, in graph theory, as vertical constraint graph.

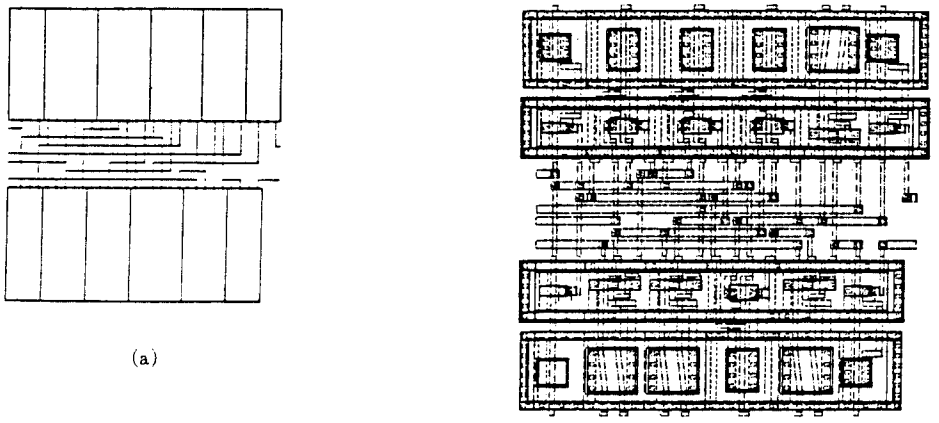
MERGE: This is the key part of the channel routing algorithm, where two sets of nets are merged. Merging of net 'a' with net 'b' implies that the horizontal segments of net 'a' and 'b' are to be placed on the same horizontal track. Vertical constraint graph and zone representation are updated after each merging operation. The purpose here is to minimize the length of the longest path after all possible merging operation in vertical constraint graph. The nets merged for the M/S D f/f considered here are shown in Fig 8 (b).

V. Run Example and Concluding Remarks

Fig. 9(a), (b) show the output result of program NETPLOT and NETCIF respectively, for D-FF with set/reset shown in Fig. 3. Fig.10 shows another example for BCD to Decimal Decoder.

Table 2 shows the number of tracks and maximum densities for the case of Fig. 9 and Fig.10.

As can be demonstrated by the fact that the number of tracks is equal to the maximum

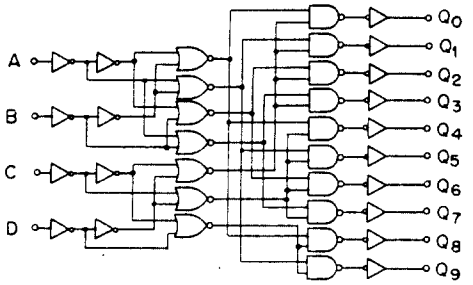


(a) (b)

Fig. 9. (a) Schematic layout result from NETPLOT for D-FF with set/ reset.
(b) Physical layout result from NETCIF for D-FF with set/ reset.

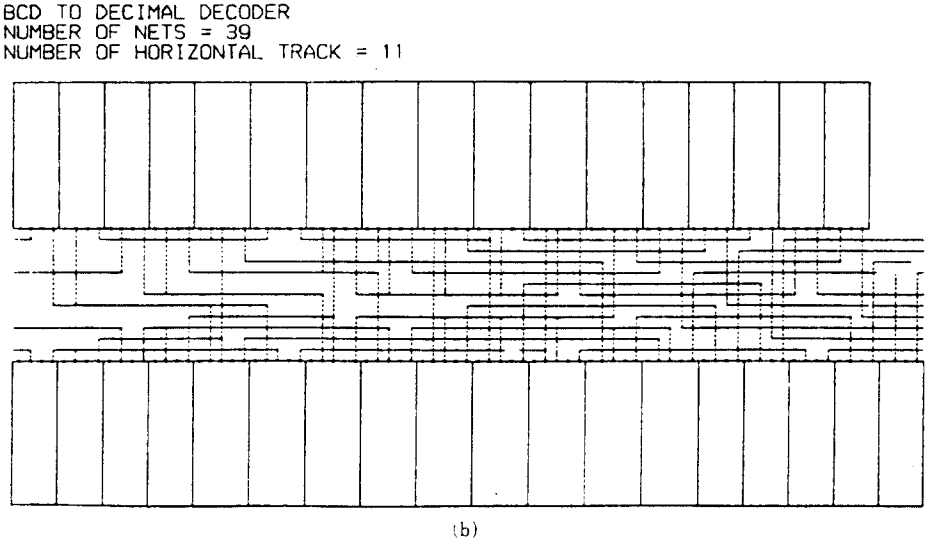
Table 2.

	# of track	maximum density
D-FF	7	7
BCD to DECIMAL	11	11



(a)

generally results near-optimal solution for most practical problems. In conclusion, a two-row horizontal wiring density for both cases, we



(b)

Fig. 10. (a) Circuit schematic for BCD to decimal decoder.
(b) Schematic layout result from NETPLOT for BCD to decimal decoder.

could conclude that the program, CROUT, channel routing program, CROUT, was written in Pascal and run successfully on VAX 11/780, in connection with various data files and graphic hardware and software tools. This program can be used for reduced gate array routing given the feed-through cell assignment as well as standard cell routing. The cell placement is, itself, worthy of another effort, and was assumed, in this paper, to have been achieved by some means. We did not include the optional doglegging to resolve the cyclic conflict, which merits further study. The routing algorithm adopted in program, CROUT, was based on what was already proposed by Yoshimura et. al. Various input data including netlist data, placement data, component library and cell library were used to specify the requirement for electrical connection and physical layout. The result of the routing was drawn on Tektronix terminal or Calcomp color plotter using various in-house graphic programs. It is believed that the proposed routing system can be extended to handle full chip routing by adding global routing features.

Acknowledgement

The authors greatly appreciate the financial support from the Korea Science and Engineering Foundation and the Ministry of Science and Technology.

References

[1] G.Persky, D.N. Deustch and D.G. Schweikert, "LTX-A Mini-Computer Based System for Automatic LSI Layout. J. Design Automation and Fault Tolerant Computing, pp. 217-255, May, 1977.

[2] C. Mead, L. Conway, *Introduction to VLSI Systems.* Addison-Wesley, 1980.

[3] Eiji Kawamoto, *Schematic Entry System with a Query Feature.* Proc. Design Automation Conference, pp. 503-506, 1982.

[4] T. Yoshimura and E.S. Kuh, *Efficient Algorithms for Channel Routing.* IEEE Trans. on Computer-Aided Design of Integrated Circuit and Systems, pp. 25-35, 1982.