Technical Memorandum

SCTM 201-60(24)

AN AID TO PRINTED CIRCUIT LAYOUTS USING THE IBM-704 COMPUTER

THIS IS A *SANDIA CORPORATION* WORKING PAPER AND IS INTENDED PRIMARILY FOR INTERNAL DISTRIBUTION; THE IDEAS EXPRESSED HEREIN DO NOT NECESSARILY REFLECT THE OPINION OF THE CORPORATION.

SCTM 201-60(24)

AN AID TO PRINTED CIRCUIT LAYOUTS USING THE IBM-704 COMPUTER

S. D. Stearns, 2421
R. A. O'Connell, 2423-1

ABSTRACT

This memorandum presents the possible applications and instructions for using REDCROSS, an IBM-704 computer program. This program was prepared for use in printed circuit layout to lower the number of man-hours normally required to reduce a circuit to its optimum form. This document does not claim that circuit layouts can now be handled completely by a computer program, but that assistance in selecting a feasible layout is available, thereby reducing the cut-and-try methods presently required.

Other possible applications of this program will undoubtedly come to mind as more persons become familiar with it.

Case No. 13.614.10 June 1960

Printed in USA. Price $0. 50. Available from the Office of Technical Services, Department of Commerce Washington 25, D. C.

DISTRIBUTION:

TISE (325)
OTS (100)
E. L. Harley, 1218, Attn: R. N. Bayless
D. M. Olson, 1262
R. L. Peurifoy, 1264
G. H. Mauldin, 1314
R. P. Stromberg, 1322
A. E. Winblad, 1322-1
R. T. DePew, 1411
J. P. Shoup, 1413
J. J. Dawson, 1414
A. W. Fite, 1423
G. W. Rodgers, 1424
W. E. Boyes, 1440
N. J. Eich, 1627
L. J. Paddison, 2400
T. F. Marker, 2420
Reading File, 2421
S. D. Stearns, 2421
R. A. Richards, 2422
R. W. Roberts, 2422-1
A. C. Ellingson, 2422-2
R. A. O'Connell, 2423-1 (10)
E. L. Deeter, 2440
W. C. Hunter, 2442
L. A. Smoll, 2442-1
W. C. Kraft, 2450
P. W. Callies, 2451
A. J. Roth, 2452
G. Carli, 2452-1
R. F. Utter, 3133
E. C. Domme, 4111
D. A. Watt, 4233
G. W. DeSato, 4411
P. A. Nicovich, 4412
D. B. Kennedy, 4413
D. R. Morrison, 5126
J. H. Scott, 5232
G. L. Miller, 5243
R. L. Brin, 8110
R. K. Smeltzer, 3421-3
W. F. Carstens, 3423
R. E. Dewhurst, 8233
W. K. Cox, 3466-1 (5)

AN AID TO PRINTED CIRCUIT LAYOUTS USING THE IBM-704 COMPUTER

Introduction

REDCROSS (REDuce CROSSings) is a program for the IBM-704 digital computer. It is used in assisting in the layout of printed circuit wiring diagrams. The capabilities of REDCROSS may be described as follows:

Given: A set of up to 99 independent, closed, bounded, simply-connected regions on a surface, every pair of which is either connected by one or more line segments or is not connected.

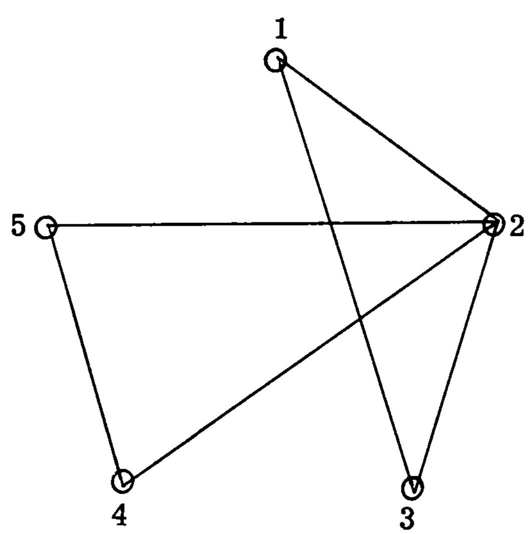
Find: An arrangement of the regions and a placement of the line segments which results in a minimum or near-minimum number of crossings of these line segments.

This is the most general way of describing what the program does. This description was used because the program is applicable to different types of circuit layout problems; viz., the regions could be connectors, plugs, component terminals, board terminals, stakes, etc., and the line segments could be wires or components.

Coding the Problem

The "given" information in the problem may be coded for the 704 as follows:

1. Assign the integers 1, 2, 3, . . . , n to the n regions.
2. Make a list of all line segments, listing each segment only once, as in the following example.



From	To
01	02
01	03
02	03
02	04
02	05
04	05

3. Using an IBM keypunch, punch each line segment in columns 1 through 4 of a standard IBM card. Punch one segment per card, e.g., 0102 would be punched in the first card, following the above example.

Running the Problem On the IBM 704

Proceed as follows:

1. (a) Assemble a card deck consisting of the program deck, followed by all of the segment cards, followed by 4 blank cards. (Program decks are available from R. A. O'Connell, Section 2423-1, Extension 3-1236.)

   (b) When computer time is requested it is necessary also to request a Printer Output tape (POP). Submit the deck to the 704 for processing.

2. Set sense switches as follows:

   Sense switches 2 and 3 represent (in binary) the number of times the subroutine will be repeated. The number of repeats needed is usually dependent on the number of regions in the problem. The following switch settings are recommended for problems of average complexity:

Number of regions	Suggested number of subroutine repeats	Sense Switch Settings SW 2	SW 3
3 to 8	4	Down	Down
8 to 11	3	Down	Up
11 to 16	2	Up	Down
16 or more	1	Up	Up

However, any combination of switch settings may be used with any problem, depending on the computer time available.

Sense switches 4, 5 and 6 control the number of cycles through the major loop of the program, and sense lights 2, 3 and 4 indicate progress by which program timing may be checked. It is generally recommended that these three switches (4, 5 and 6) be left in the down position for all problems of 16 regions or less, and that the time be noted between successive light indications. In this manner, the program time can be determined by observing the elapsed time between program start and light 4, or elapsed time between light 4 and light 3, etc. The setting of sense switch 1 down will cause the program to list on tape the best results at the completion of the current cycle, and then stop. This switch may be set at any time during the program, and its use is advised when the time required to complete the remaining cycle of the program will probably exceed the remaining time scheduled on the computer. A typical problem containing 8 vertices will take approximately 10 minutes to complete. For problems of more than 16 vertices it is recommended that all switches be left in the up position because of the time normally required to complete a problem of this size.

3. Upon completion of the program, the results will be copied on the POP tape. The card deck can then be run out and removed from the card reader and retained. The segment cards may be removed and the program deck re-used for other problems. Interpretation of the POP tape will be handled by the computer operating organization, and results will be received in printed form as depicted in results of problem No. 1, which follows:

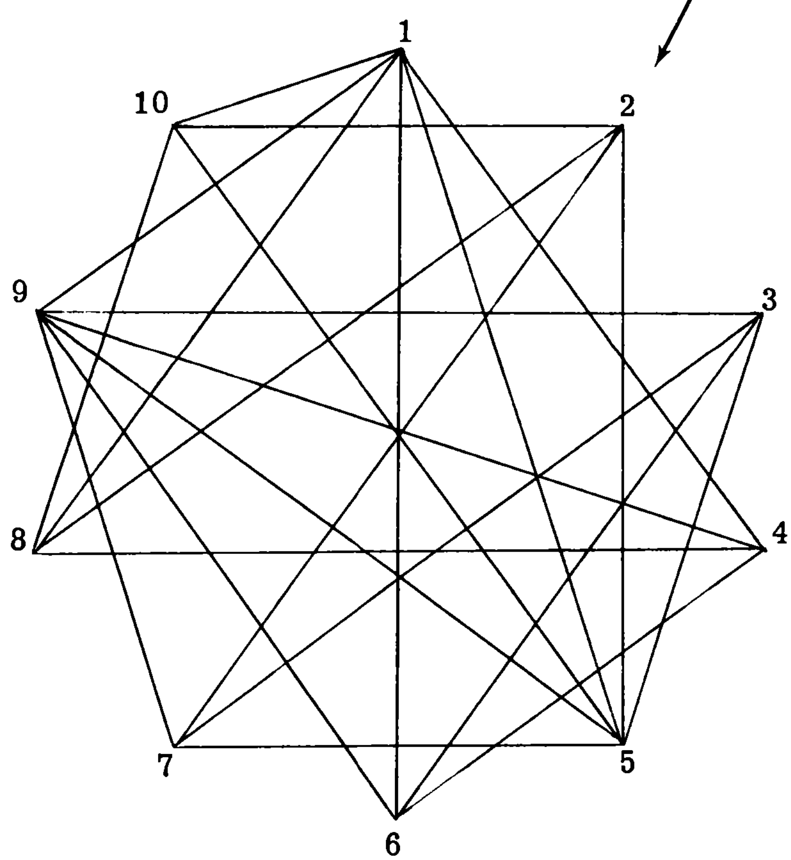
   The following are examples of actual problems that have been run, using the IBM-704 REDCROSS program.

Problem No. 1

Number of vertices (regions) = 10

Number of line segments = 23

Pictorial representation of the problem (Number of crossings = approximately 85).



Coded listing of line segments

From	To	
01	04	on card No. 1,
01	05	on card No. 2,
01	06	etc.
01	08	
01	09	
01	10	
02	05	
02	07	
02	08	
02	10	
03	05	
03	06	
03	07	
03	09	
04	06	
04	08	
04	09	
05	07	
05	09	
05	10	
06	09	
07	09	
08	10	

(IBM cards, columns 1 through 4, are punched from this list)

Results of Problem No. 1

The following figures show how the IBM-704 results are printed:

Original	Internal	External
1 - 4	10 - 1	7 - 4
1 - 5	10 - 6	1 - 8
1 - 6	10 - 2	6 - 4
1 - 8	10 - 3	10 - 8
1 - 9	10 - 9	1 - 3
1 - 10	7 - 6	0 - 0
2 - 5	7 - 8	0 - 0
2 - 7	7 - 9	0 - 0
2 - 8	5 - 6	0 - 0
2 - 10	5 - 2	0 - 0
3 - 5	5 - 4	0 - 0
3 - 6	5 - 3	0 - 0
3 - 7	1 - 2	0 - 0
3 - 9	6 - 3	0 - 0
4 - 6	6 - 9	0 - 0
4 - 8	2 - 3	0 - 0
4 - 9	4 - 3	0 - 0
5 - 7	8 - 9	0 - 0
5 - 9	0 - 0	0 - 0
5 - 10	0 - 0	0 - 0
6 - 9	0 - 0	0 - 0
7 - 9	0 - 0	0 - 0
8 - 10	0 - 0	0 - 0

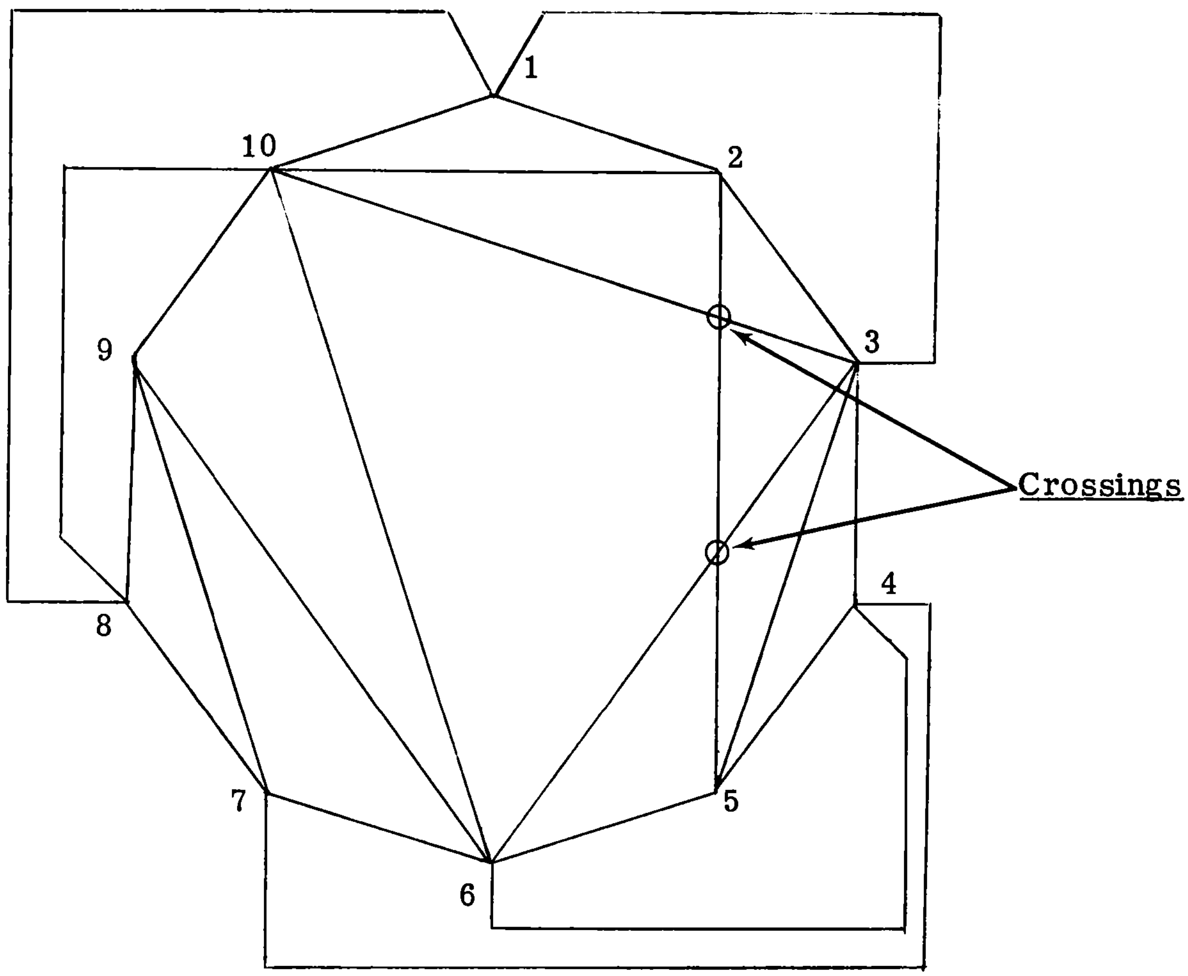
1	2	3	4	5	6	7	8	9	10
4	6	9	7	3	5	2	8	10	1

(An error in card-punching may be discovered by comparing the column headed "Original" with the coded listing of line segments from which the cards were punched.)

The vertex positions were rearranged by the program, as shown in the last two rows of figures, where the top row indicates the new arrangement and the lower row indicates the previous position of each vertex.

Although problem No. 1 contained only 10 vertices, up to 99 may be used. For problems having more than 20 vertices, the top row of figures in the printer output will not be continued, so that the third row of numbers would represent the old arrangement of vertices 21 through 40, and the 4th row would represent 41 through 60, etc.

If these results are diagramed, the pattern can now be viewed.



As can be seen, two crossings remain and no other arrangement can reduce this problem to zero crossings. However, in printed circuit layout, this represents only one side of a board: thus, it is now easily seen that by placing one line segment (2 to 5) on the reverse side of the board, a feasible printed circuit layout with no crossings is available. This particular problem took 6-1/2 minutes of computer time, including loading program, loading problem, computation and print-out.

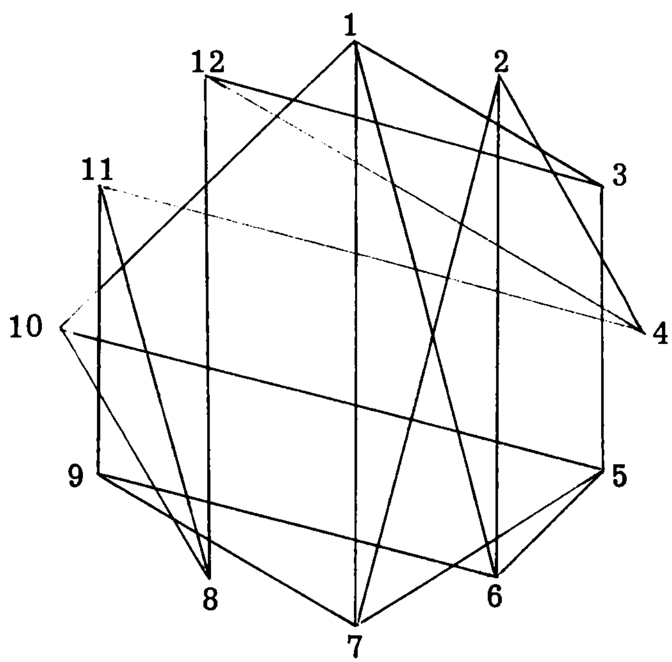
Problem No. 2

Number of vertices (regions) = 12

Number of line segments = 20

Number of crossings = approximately 45

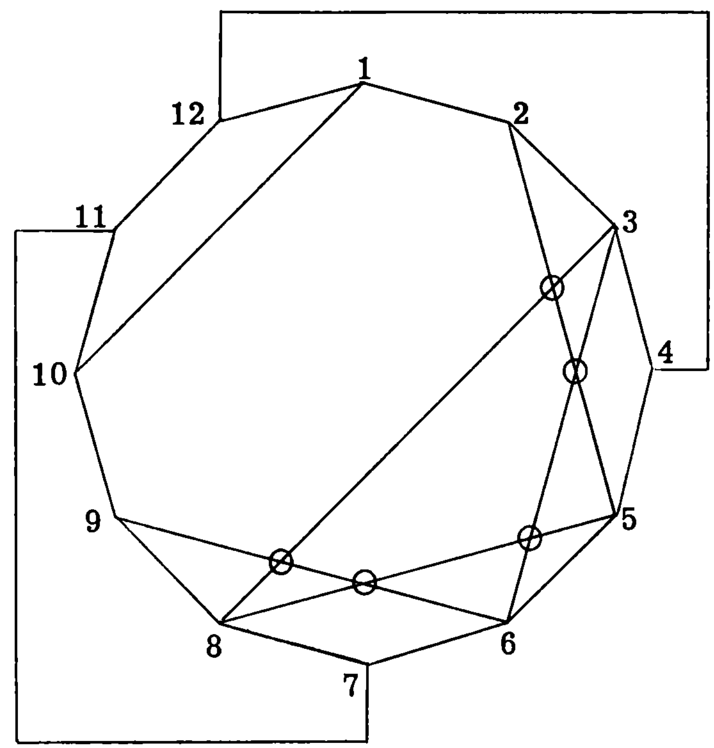
The Problem



The Positional Interchanges

1	12
2	3
3	5
4	10
5	1
6	7
7	9
8	6
9	2
10	4
11	11
12	8

The Result



The result indicates a reduction to 5 crossings; however, this problem was run for only 8-1/2 minutes. It is known about this problem that an arrangement giving 3 crossings is possible, and, by placing 3 line segments on the reverse side of a printed circuit board, it is a feasible layout with no crossings. Although the crossings have been reduced by a large amount, this problem could now be recoded and run for perhaps an additional 5 minutes and the crossings would then be minimized. Actually, there is no way to determine when the crossings are minimized unless they go to zero. Since this is often impossible, the next best thing is to run the problem only for a reasonable time and then note the results to see if sufficient progress has been made.

From the above examples, it can be seen that the computer time required cannot be easily predicted. To make the most efficient use of computer time, it is usually better to run the more complex problems for a shorter period (5 to 15 minutes), then recode and continue the problems, if necessary, after they have been examined manually. Normally, there is little or no time advantage in attempting to reduce the number of crossings manually before coding the problem, since the computer time required is more dependent on (number of line segments) times (number of regions) than on the number of crossings. The program is so written that all line segments are coded as internal connections, and the choice of external connections is made during the operation. Another condition established in the program is that no crossings are allowed in the external connections.

Upon completion of a problem, only the arrangement of the vertices is established. Layout to fit the needs of the particular application can now be made and it can easily be established how many surfaces are required to eliminate all crossings, if this is necessary.

This type of work is considered production on the 704 computer. At present, no more than 15 minutes of daytime computer production may be scheduled at a time. Work of this type should be handled by the night shift and may be requested on a "704 Service Request" form. This form and further information on computer scheduling are available from Division 5243, Extension 3-4231.

Some Practical Considerations

The primary work of this program was in connection with the layout of interconnecting wiring in places such as interconnecting boxes, where few, if any, components are involved. When this program is used for schematic layout, no problem exists until components having more than two terminals are encountered, since a component of this type may be considered as a line segment. When three-terminal components, such as most transistors, are encountered, then these items may be considered as regions, provided that the component connections are flexible so that their connecting arrangement can be varied. This, again, is the case for most transistors. However, since this program has no provision for fixing the location of any one point with relation to another point on the boundary of a given region, items with three or more nonflexible leads or connections cannot be successfully considered.

This rule is not as strict as it might at first seem. For example, consider a connector with any number of terminals. This connector may be considered as a single region in the problem, providing the pin numbers are not assigned before coding the problem, and can be assigned at random upon completion of the problem and based upon the results achieved.

This method allows all points on a multiple terminal component to be considered as one, thereby giving the computer program no opportunity to interchange them with other points or components.

In the case where schematics are to be coded, line numbers must also be assigned and maintained during the coding process so that their identity will be known at the completion of the problem. Once the identity of the lines is known along with the points of termination, these lines may be considered as two terminal components, such as resistors, capacitors, etc. The line number is not punched into the IBM cards. An example of identifying components follows:

R1 is line 1 and connects point 1 to point 7

R2 = line 2 and connects point 7 to point 22

C4 = line 29 and connects point 28 to point 2

The points of termination, such as 1, 2, 7, 22, and 28, above, are punched into IBM cards as vertices, but now it is known that line segment (7 to 22), for example, is not just a line but is resistor R2 of the schematic being coded.

Terminations of components, connectors, etc., are assigned region numbers directly, and these numbers become the vertices of a problem. Therefore, these numbers are punched into IBM cards and used in the program. It is advisable to mark these points on the schematic as they are assigned.