研　究　報　報

UDC 681.3.06

# AUTOMATIC FLOWCHARTING

自動フローチャーティング

by Seiichiro YAMAMOTO* and Kusuo YAMAGUCHI**

山 本 征一郎・山 口 楠 雄

## 1. Introduction

A flowchart is the simplest and the clearest means to show the procedure of information processing. The flowchart is aimed at the final materials of coding, the means of debugging or the documentation of program. To make a flowchart is, however, very boring job, and needs much effort. Especially, a detailed flowchart which corresponds closely to the program usually has a lot of quantity and needs frequent corrections. Therefore, if such a flowchart can be obtained easily as a program list can, the labor of making program would be much reduced. For such a reason, the researches concerning automatic flowcharting has been performed for more than a decade. Some of these researches are aimed at analyzing programs, and others have the purpose to get flowchart actually.

The system of this article consists of following methods. The first is to analyze a program from two viewpoints, that is function and control-flow, and to compose blocks and groups. The second is the method of sequential arrangement. This method largely reduces memory capacity. Flow lines are drawn after the drawing of all groups.

The aim of this system is to get an easy to see flowchart in the simple way. In many other systems a lot of memory capacity is necessary and to change the size of flowchart symbols is not easy, because they use a line printer as an output machine, and they draw a flowchart after the arrangement of all blocks. This system uses X-Y plotter, it improves these defects and produces flowcharts of good quality.

By the present time, this automatic flowcharting system has been applied for assembler language, the reason is that the merit of flowcharting in assembler is considered larger than in other languages. The system, however, can be applied for compiler by changing its method of analyzing program slightly.

## 2. The system of flowcharting

What are important in flowcharting of computer program are to display the processing meanings by proper symbols, to rearrange one dimensional program to the easy understandable two dimensional chart and to show the relations of the control flows of program clearly. The merit of flowcharting depends on how to do above things.

### (1) The method of block composing

A block is an unit or a symbol of program on flowchart. It is necessary to compose a block or to collect instructions into a block when flowchart is made from the program of assembler language. For composing a block it is considered to take out the series of instructions which performs one function out of long series of instructions by the rules stated below.

1) One block is to be separated by the instruction which is prior to the instruction with label or to the instruction to which a branch flow comes in.

2) As to a shift instruction, a set instruction of astatus flip-flop and a control instruction, one of these instructions is to make one block.

3) As to a arithetic instruction, the series of instructions from LOAD instruction to STORE instruction is to make one block.

* Former Graduate Student, Univ. of Tokyo
** Dept. of Electrical Engineering and Electronics, Inst. of Industrial Science, Univ. of Tokyo.

4) The SET instructions of index register or that LOAD instructions are to be collected into one block as far as the same kind of instructions continue.

5) The same kind of conditional branch instructions are to be collected into one block up to three branches.

6) In case that above rules are incompatible each other, the preceding rule is prior to others.

The block composing method of arithmetic instructions under these rules are shown in Fig. 1. Branch data (BRD) contains 16 word of data for each branch instruction. Block data (BLD) contains the data which are prepared at block composing, and these data are the kind and the size of flowchart symbol, the start position of drawing, present pen position, condition of branch and the characters to be written in the symbol. BLD occupies the common memory of 16 words to each consecutive block.

### (2) Grouping

A set of blocks is to be an unit of arrangement and called a group. The grouping is made by program analysis concerning branch instructions. The algorithm of grouping is that an unconditional branch instruction is to terminate a group unless it follows directly a skip instruction or it is an exit from the midle of a subroutine. A conditional branch instruction doesn't terminate a group.

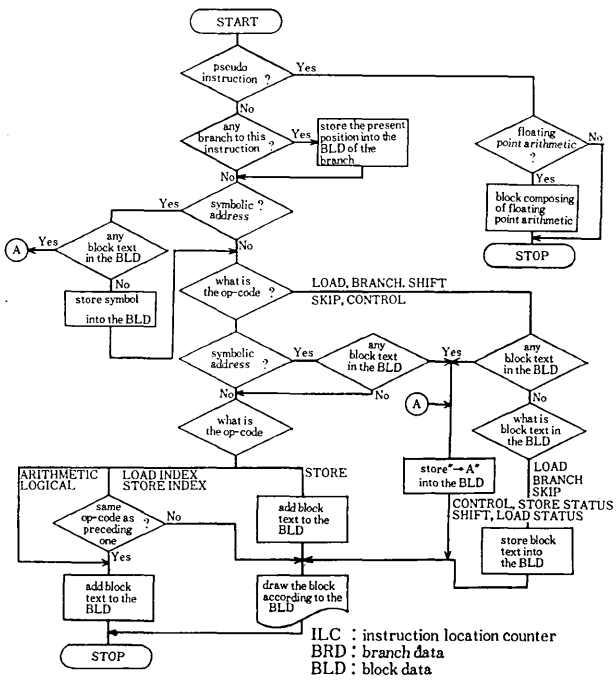


Fig. 1 Block composing procedure for arithmetic instructure.

### (3) The method of arrangement

How to arrange groups on space is one of the most important factors which affect the understandability and the beauty of a flowchart. However, the high quality of a flowchart and the simplicity of the flowcharting program are usually incompatible. In this system, the sequential method described below is employed to satisfy both conditions as well as possible.

1) Space is partitioned into four domains as shown in Fig. 2, and drawing starts at the second domain, that is domain 3.

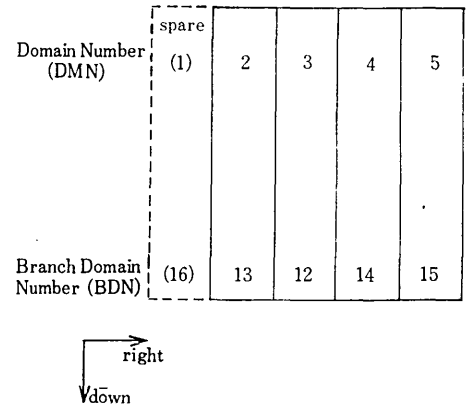


Fig. 2 Partitions of chart space, and their Domain Number and Branch Domain Number.

2) The direction of drawing is downward except flow lines.

3) The group to be drawn and its starting location are to be determined sequentially after the drawing of one group is finished.

4) The group to be drawn next is determined by the branch domain number.

5) Domain number is to be given to the group of which domain to be drawn has been decided.

6) For choosing the group to be drawn,

the group to which domain number have been given is searched first, the group to which branch domain number have been given is searched next.

7) If more than two groups have been given branch domain unmber, the group which has branch from the highest location is to be chosen.

8) A branch domain number itself does not mean to which domain, left or right, the branch is to be done. The choice of the domain to which a branch is to go depends on whether that group has branch to another group, and on the relation the location of other groups already drawn.

9) The start position of drawing a group is to be the same height of the branch point to it of the adjoing group, unless it overlaps. If more than two branches come down to a group, the group is to be drawn at the heighest position of the branches which do not overlap. If all branches overlap, the group is to be shifted down to the first free space.

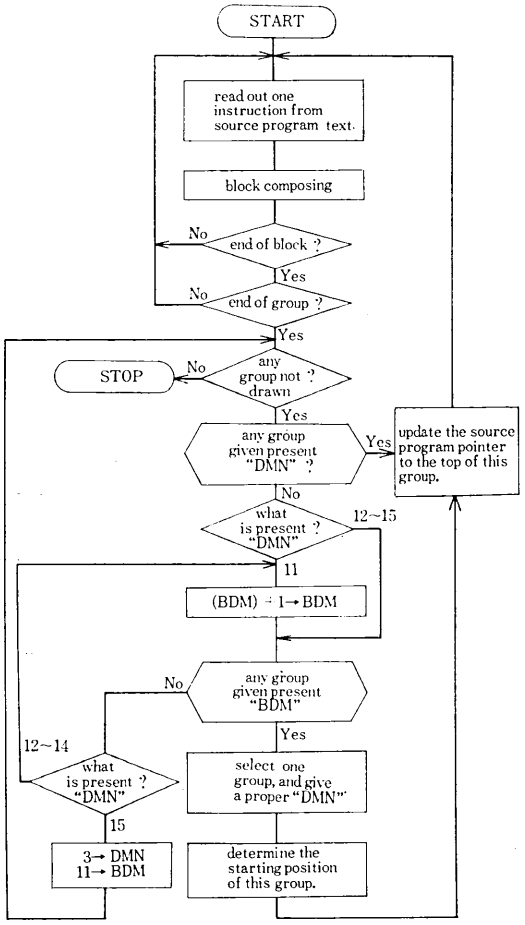


Fig. 3 Algorithm for arrangement

### (4) Flow line

The method of drawing flow line is composed of the rules stated below.

1) Flow lines are to be drawn according to the branch data after all groups drawn.

2) The spaces for three lines are in the both sides of every domain, and one of them is to be selected when a flow line is to be drawn. Therefore, a flow line is composed of one vertical line and two horizontal lines. The spaces for flow lines are given the number of 6 to 31.

3) A flow line is to be drawn when the branch domain of destination is the same one or adjoining one. Connectors are to be employed the branch domain of destination is farther than these domains.

4) The flow line which goes to the adjoining domain is to be given the line number first, and one which goes to the same domain next. The flow line which has shorter loop is to be given priority.

5) There is two cases for giving the line number according to the branch destination

(a) In case that the adjoining domain is the

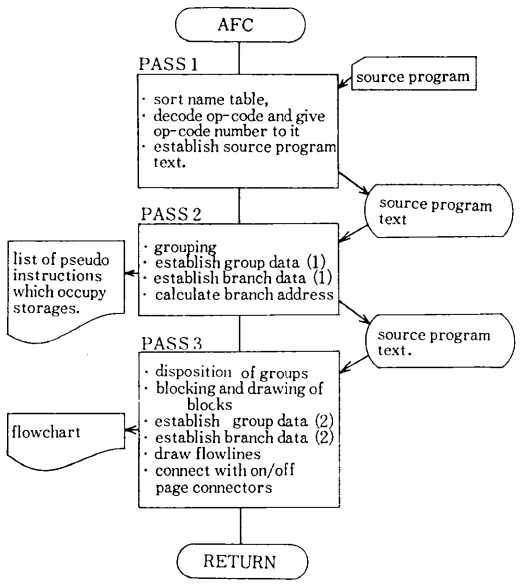


Fig. 4 The Block diagram of automatic flowcharting program

branch destination, a flow line is to be given one of the numbers of four middle lines of six lines between the two domains unless the line overlaps.

(b) In case that the same domain is the branch destination, a flow line is to be given one of the numbers of eight lines which are in the both sides of the domain unless it overlaps.

If there is no line number which satisfies the conditions in (a) or (b), connectors are to be used to connect those two groups.

## 3. Automatic flowcharting program

This automatic flowcharting program adopts three pass system which reads source program three times as shown in Fig. 4.

In PASS 1 as shown in Fig. 5, source program is read from cards, and labels are registered in label table. Then, an instruction are decoded, and the corresponding instruction number are given to it, and the present instruction location counter (ILC) and the estimated instruction length (SL) are written into source program text. Such process is repeated every time a card is read until END comes up.

In PASS 2 as shown in Fig. 6, an instruction of the source program is read out, and it is checked whether it is a branch instruction. If it is not a branch instruction, the process moves to the next. If it is a branch instruction, it is checked whether that group can be terminated, and branch data is made. During this process, the address of branch destination, the condition of branch and the address of the branch data of this instruction are added to the source program text. These processes continue until END comes up.

In PASS 3 as shown in Fig. 7, the first group

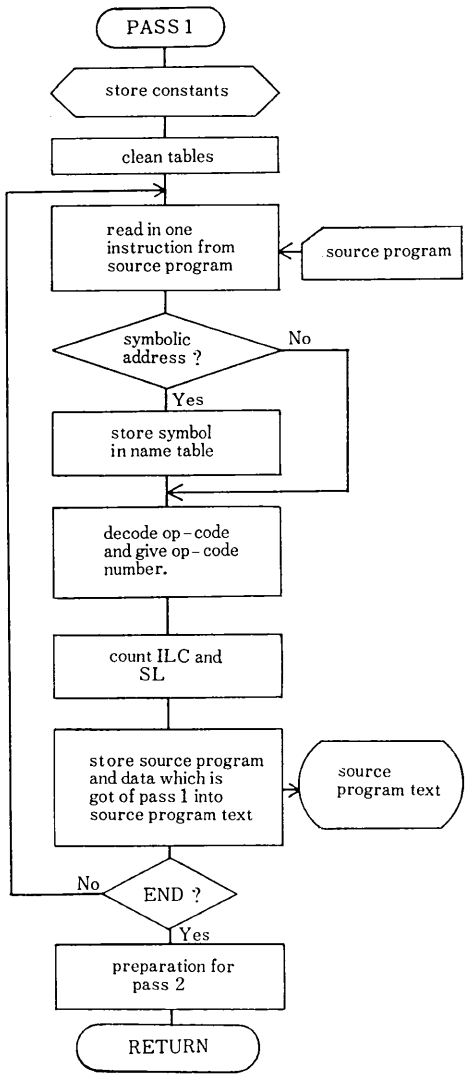


Fig. 5 The flow chart of PASS 1

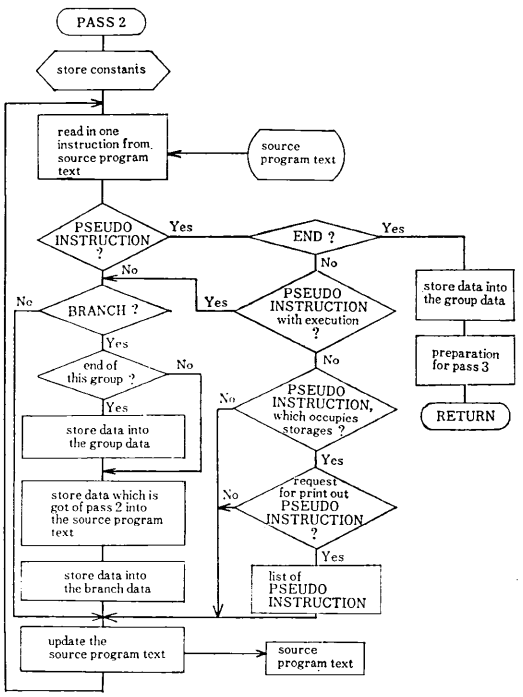


Fig. 6 The flow chart of PASS 2

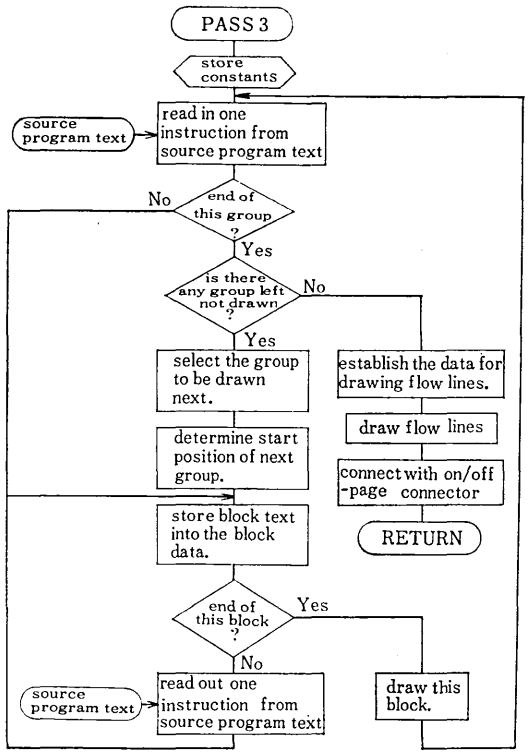


Fig. 7 The flow chart of PASS 3

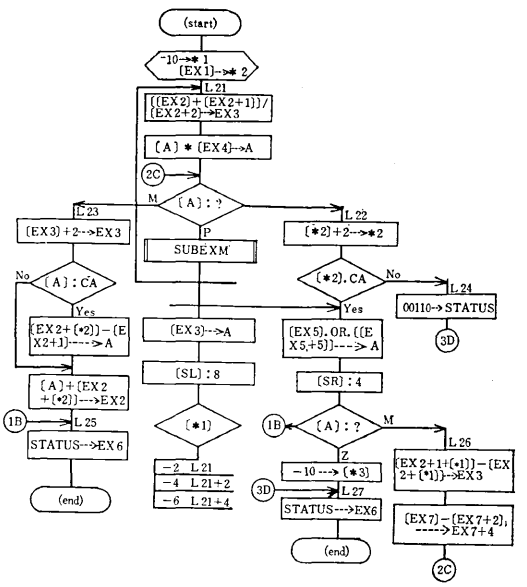


Fig. 8 An example of flow chart drawn by AFC program

is read out and blocks are made from it. A block is drawn on flow chart soon after it is composed. After the first group is drawn, next group and its start location are determined and the group is drawn on the sheet. After all the groups are drawn in such a way, the process to draw flow lines starts. The location data necessary to draw flow lines starts. The location data necessary to draw flow lines are written in branch data when block composing is made. Line number and branch direction are determined when a flow line is drawn. These processes continue until all the flow lines are drawn. Then, PASS 3 is finished and the flowchart is completed.

## 4. Conclusion

Automatic flowcharting experiments by this program have been performed on various programs, and almost satisfactory results have been obtained. But, rarely long vertical flowcharts without proper arrangement to horizontal have shown up. The cause for this is that the system employs the sequential method for the symplicity of the flowcharting program and the method does not divide a group by conditional branch for improvement of correspondence to the program. In addition, composing the blocks of a call subroutine instruction which has more than two exits and of the instruction of which branch destination is decided during operation are solved by using coments. The memory capacity which is used for this automatic flowcharting program is about 24 K-words (1 word =2 bytes) except source program text. The time to draw a flowchart of a 50 steps program is from 7 to 10 minutes by computer FACOM 270-30 including drawing time of X-Y recorder. Actual computing time is about 1 minute which could be a reasonable value for practical use. This automatic flowcharting program is available in the computer room of Institute of Industrial Science, however ithas slight defects which areto be improved.

## Acknowledgement

The authors wishes to express their appreciation to Mr. T. Shimada in the Third Department of Institute of Industrial Science, University of Tokyo, and Mr. M. Sakurai who was in the same depart-

研究速報

ment for their kind help to make the flowcharting program, and to the members of Professor M. Watanabe's laboratory for their useful advice.

(Manuscript received Sept. 17, 1971)

### Biblography

1) Richard M. Karp: A Note on the Application of Graph Theory to Digital Computer Programming, Information and control, Vol. 3. No. 2 (1960), pp. 179-190
2) Edward A. Voorhees: Algebraic Formulation of Flow Diagrams, Communications of the Association for Computing Machinary, Vol. 1, No. 6 (1958), pp. 4-8
3) Lee Krider: Flow Analysis Algorithm, Journal of the Association for Computing Machinary, Vol. 11, No. 4 (1964), pp. 429-436
4) Lois M. Haibt: A program to Draw Multilevel Flow Charts, Proceedings of the Western Joint Computer Conference, 1959, pp. 131-137
5) Mamoru Maekawa: Automatic Flowcharting, Information processing, Vol. 9, No. 3 (1968), pp. 131-136
6) IBM Application Program, System/360 Flowchart (360A-SE-22S) User's Manual
7) S. Yamamoto & K. Yamaguchi: Automatic Flowcharting, Information Processing, Vol. 11, No. 12, (1970) pp. 711-720

## 次号予告 (12月号)

**研究解説**



**研究速報**



**研究室紹介**



## 正誤表 (10月号)

| ページ | 段 | 行 | 種別 | 正 | 誤 |
|---|---|---|---|---|---|
| 23 | 右 | 17 | 本文 | 丸安研究室 | 丸安研空室 |
| 31 | 左 | 下11 | 〃 | $a_0S^3+{}_1S^2+a_2S+a_3=0$ | $a_0S^3+a_1S+a_2S+a_3=0$ |