M.E.T.U. Journal of the Faculty of Architecture
Volume 5, Number 2, Fall 1979

A COMPUTER GRAPHIC PROGRAM FOR ARCHITECTURAL DRAFTING[1]

Bülent ÖZGÜÇ
Mustafa PULTAR

Received May, 1979

COMPUTER GRAPHICS

The application, in the past decade and a half, of computer based data processing to graphic data has given rise to a field of activity known as computer graphics.[2] Numerous and fundamentally different disciplines such as navigation, biochemistry, mining, medicine, etc. have been influenced to very great extents by it. The potential and actual effect of computer graphics on design, especially architectural design, has manifested itself in three principal directions.

The first, and more obvious, of these is that the time consuming, expensive and routine activity of drafting may be turned over, within certain limits, to the computer. In industrialized economies, firms offering the use of computer graphics facilities appear to undertake increasingly more of the design, or at least drafting jobs, of the bigger and complicated buildings. The firm of Saphier, Lerner and Schindler-Environetics, Inc. of New York, for example, has been able to draft the total architectural drawings of the Sears Tower by using a computer based drafting system. The same is also valid for structural, electrical, mechanical and other system drawings.

A second use of computer graphics in architectural design is in the generation of design alternatives and the production of statistical data therefrom for purpose of

1. This article is based on a Master of Architecture thesis written by the first author and supervised by the latter. B.ÖZGÜÇ, "An Interactive Algorithm for Architectural Drafting." Unpublished Master of Architecture Thesis, Middle East Technical University, Ankara 1975.

2. See, for example, W.M. NEWMAN and R.F. SPROULL, *Principles of Interactive Computer Graphics*, New York: McGraw Hill Book Company, 1973, and D.P. GREENBERG, Computer Graphics in Architecture, *Scientific American*, v.230, n.5, May 1974.

evaluation. Especially in a mode of operation where the designer is interacting with the computer, he can produce alternatives continuously, can evaluate these either on the basis of data produced by the computer from the design or other data which the designer produces himself and can arrive at a final solution which he deems satisfactory. Such a mode is preferable over one of total automation of the generation of design alternatives, which presents several difficult problems in itself.

A third significant use of computer graphics in architectural design is in the handling of difficult drafting problems where the use of other techniques is not feasible or even possible. Examples are the perspective projection of complex constructs and sciagraphic problems.

The present article deals with the fundamentals, development and presentation of software capable of handling the first two of the computer graphics problems outlined above.[3] The description is given in the context of planar architectural drafting, but the software can easily accomodate other contexts simply by a change in the graphical vocabulary. In fact, this facility forms one of the fundaments of the design of the software.

3. The description of a similar program is given in GINGER: Graphics Interaction with General Elements, *ABACUS Occasional Paper*, n.65, University of Strathclyde, Glasgow, March 1976.

FUNDAMENTAL CONSIDERATIONS

Software which is capable of adaption to both of the computer graphics modes outlined in the preceeding section should incorporate the following features:

1. Operation with a strictly graphical (geometric and/or symbolic) vocabulary. If so desired, the operation should be independent of the metric properties of the graphical vocabulary.
2. Facility of changing the content of the graphical vocabulary at will without fundamental alterations of the software. This feature will form the basis of the adaptability of the software to various and essentially different drafting problems such as architectural, structural, electrical drafting for buildings, site layout, plumbing or circuit design, textile design, graphic design, etc.
3. Facility of geometric transformations on graphic content such as transformations of location, orientation, size, affine distortion, etc.
4. Interactive mode of operation. This feature will allow the designer or draftsman to alter designs easily at will without significant loss of time and material. An interactive mode of operation will imply independent monitoring and hardcopy availability.
5. Production of statistical data on designs. This feature will allow the designer to make decisions and especially changes on the design.

Such features point to a user-computer system where the function of the two components may be differentiated as follows. The designer

a) defines a vocabulary consisting of planar graphical elements, operable by formal transformations -this feature may be an integral part of the software-
b) generates designs in the form of planar layouts composed of the repetitive use of the graphical vocabulary subjected to transformations,
c) evaluates the design and makes alterations as necessary on the basis of decision variables produced either by himself or the computer.

The computer, on the other hand,

a) performs all geometric and metric transformations on the given or pre-defined graphical vocabulary,
b) produces all required images or designs in the form of temporary or permanent records, and
c) produces and computes data on decision variables as desired.

The ease at which the user-computer system will operate is a function of the input and output devices available at a particular computer installation. The ideal layout is an electronic tablet[4] for input, a refreshing cathode ray tube (CRT)[5] for monitoring and a digital line plotter for hardcopy outputs.

4. For information on electronic tablets as input devices, see L. GALLENSON, *A Graphics Tablet Display for Use Under Timesharing*, Washington, D.C.: Thompson Books, 1967.

5. The preference of a refreshing CRT to a storage one is basically due to the fact that if a mistake is made on one part of the picture, it can be deleted immediately and corrected at the point where the mistake occured on a refreshing tube. Even for local corrections, however, the whole picture has to be erased and redrawn on a storage tube.

How convenient the coupled use of electronic tablets and refreshing CRTs for graphical input and output is obvious. In general, an item of the graphical vocabulary can be drawn on the tablet and given a name; this suffices for its specification. Each time its use is necessary, it is recalled by name, subjected to the desired transformations and appears on the CRT. When the design is complete along with recalls of other items, the layout on the CRT can be transferred to real metric scale and routed to a line plotter for a hardcopy.

In the design of the user-computer system, user requirements and computer capabilities have to be thought of simultaneously. In general, the easier things are made for the user, the more the energy that the computer has to spend. In certain cases, trade-offs have to be made. Within the hardware itself, as far as data structuring is concerned, trade-offs have to be made between computer time and storage space in its memory.

PLAN: AN INTERACTIVE DRAFTING PROGRAM

In light of the considerations above, the program plan was prepared[6] for operation on the IBM 370 installation of the Uni-Coll Corporation (University of Pennsylvania) with a Calcomp 763 digital plotter. PLAN is essentially oriented towards architectural drafting with a graphical

6. B. ÖZGÜÇ, "An Interactive Algorithm for Architectural Drafting", Unpublished Master of Architecture Thesis, Middle East Technical University, Ankara, 1975.

vocabulary consisting of templates of structural elements, furniture and fixtures. The program is capable of translating, rotating, and scaling the templates and drawing them with different line thicknesses on a digital plotter or with different intensities on a CRT monitor.

The graphical vocabulary of PLAN is shown in Figure 1 along with the template names (commands) for each item as listed in Table 1. This vocabulary is contained internally in the PLAN program; each item is digitized and plotted by subroutines of PLAN. However, by changing subroutines and commands only in the input, the vocabulary may be altered easily so that PLAN may be adapted to other drafting purposes.

Structural Templates (may be given only in the BEG mode)		
Command		Subroutine
ELV	Elevators	MAIN
STC	Staircase	MAIN
COL	Rectangular Column	MAIN
IRR	Irregular Structure	MAIN
Furnishing Templates (may be given only in the FUR mode)		
Command		Subroutine
REC	Rectangle	RECTNG
CIR	Circle or arc	CIRCLE CIRSEG
WN1	Window with mullion	WINDO1
WN3	Single window	WINDO3
DRN	Door open 90°	DOOR
PR2	Interior partition	PART2
SEC	Secretarial layout	OFFICE
ARM	Armchair or couch	ACHAIR
TEL	Television	TLVSN
SHO	Shower	SHOWER
BAT	Bathtub	BATH
URI	Urinal	URINAL
PLU	Electrical outlet	PLUG
OUT	Special purpose outlet	OUTLET
PHO	Telephone outlet	TRIANG
TRI	Triangle	TRIANG
ELL	Ellipse	ELLPS
WN2	Window with regular divisions	WINDO2
DRF	Door open 45°	DOOR
PR1	Interior partition	RECTNG
OFF	Office layout	OFFICE
CHA	Chair	CHAIR
BED	Bed	BED
PIA	Grand piano	PIANO
SIN	Sink	SINK
BOW	Commodoe	IT
LHT	Lighting fixture	LIGHT
FLO	Floor outlet	FLOLET
TRE	Tree	TREE
TAB	Table	RECTING

Table 1. Templates of Plan

7. If real display files of graphic installations are thought to be buffers, those which are included in PLAN are better called "pseudo-buffers". Usually for graphic uses, immense storage area is required for keeping the coordinates of the digitized picture. This is where a trade-off is necessary. Instead of storing the digitized picture, it is possible to keep the input specifications and re-execute them whenever plotting of that pseudo-buffer is required. This is very effective when dealing with limited memories.

A schematic diagram of the user-computer system operating with PLAN is shown in Figure 2. The user specifies input through the terminal or graphic input devices; this input is either in the form of commands related to the choice of pseudo-buffers[7] for output and output commands or in the form of item specifications along with information on size, location, orientetion, line thickness and intensity related to each item.

PLAN uses two pseudo-buffers, where one is reserved for the structural layout and the other one for the remaining templates. In the first buffer, the user may specify columns, walls, irregular structures, staircases and elevators. Into the second buffer, four basic geometric shapes (triangle, rectangle, circle and ellipse) and twenty six architectural symbols may be fed. These pseudo-buffers can be overlaid as many times as desired or each plotted separately. One pseudo-buffer may be erased while the other one is kept intact. Additions to or deletions from the buffers may be made at any time, but the user should make sure that he is putting the correct information into the particular buffer. Otherwise, error messages are given. There are control statements in the program which make it possible to jump from one buffer to the other. At the beginning of each run, the structural layout buffer will be ready to receive commands or templates.

In order to change buffers, the command END has to be given; this sets the program ready to take further commands. Available commands are as follows:

- END Stops filling in the buffers and expects to receive commands. Only the following six statements (PRI to STO) can be given after END.
- PRI Prints on a teletype or line printer every input specification that has been given previously and contained in the last pseudo-buffer before END was specified.
- PLO Plots whatever has been fed into the display file previously on the hardcopy device.
- REP Re-executes the last buffer but does not send information to the display file for the hardcopy until PLO is specified. However, it replots the last buffer on the monitor.
- BEG Enters the program into the beginning mode and the structural layout buffer. After this command, only structural elements (Table 1), DEL or END can be specified.
- FUR Enters the program into the second (furnishings) buffer. Operation of FUR is identical to BEG; instead of the structural elements only furnishings and symbols may be fed in.
- STO Stops and terminates the program.
- DEL Can be specified either in the BEG or FUR modes. This command deletes any unwanted item from the buffers. The user points out these unwanted items by feeding in the input sequence number of the item.

SCR Can only be specified in the FUR mode. It scratches and initializes the furnishings pseudo-buffer.

ARE Can only be specified in the FUR mode. It culculates the areas of closed irregular polygons by integration.

An important convention related to output concerns the REP and PLO commands. When PLO is specified, a hardcopy is given and the plotter is advanced (for the case of drum plotters) to a clean page. When REP is specified, the last buffer in question is re-executed, plotted on the CRT monitor but not on the digital plotter. Then, without specifying a PLO command, the user can shift to a different buffer and again specify a REP command which will superimpose the new picture over the previous one. If PLO is then given, the hardcopy will contain the superposed drawing. Without specifying PLO in between, the user can scratch, fill and superpose as many pseudo-buffers as he wishes by using REP.

When filling in the buffers, each item name is followed by the attributes of the item by pointing to the location and other regions specifically denoted on the electronic tablet. When the user specifies an input, it is fed into the required pseudo-buffer and at the same time transferred into a display file for the CRT monitor and actually displayed. If there is a mistake, or the user does not like the appearence of the displayed figure, he can delete it. When the picture is completed, the information in the pseudo-buffer can be routed to the buffer of the hardcopy device, which is a tape containing digitized information. Monitoring, thus eliminates the problem of corrections on digitized information, as well as the cost of many hardcopies which are especially expensive and slow if liquid ink is used.

The relative quality of output can be seen by comparison of Figures 3 and 4, which are outputs obtained, respectively, on the CRT monitor and the digital plotter.[8] These figures are related to the output in Figure 5 which contains information on dimensioning; this facility was added in a later version of PLAN.

8. Minor differences of composition will be noted between these two figures. These differences are due to their being plotted at different stages of the run rather than their being plotted on different output devices.

Certain other internal features of PLAN such as data structure and flowcharts are given in the Appendix.

SAMPLE RUN

To illustrate the use of PLAN, we include here a sample run with the list of input statements used during the run (Table 2) and the hardcopy outputs produced. Information following the item specifications has been deliberately left out as this would entail lengthy detail on the attributes of each item.

```
01  IRR  0
02  IRR  0
03  IRR  0
04  IRR  0
05  IRR  0
06  STC  0
07  STC  0
08  COL  0
09  COL  0
10  IRR  0
11  COL  9
12  IRR  0
13  COL  0
14  ELV  0
15  ELV  0
    END
    PRI      (Produces, on teletype, list of input so far)
    PLO      (Produces Figure 6)
    REP
    PLO      (Produces Figure 6 on a new page)
    FUR
01  WN1  0
02  WN2  1
03  WN1  0
04  WN1  0
05  WN1  0
06  WN1  0
07  WN1
08  WN1  0
    END
    PLO      (Produces Figure 7)
    FUR
09  PR2  0
10  PR2  0
11  PR2  0
12  DRN  0
13  DRN  0
14  URI  0
15  URI  0
16  PR2  0
17  DRN  0
18  PR2  0
19  DRN  0
20  PR2  0
21  SIN  0
22  SIN  0
23  BOW  0
24  BOW  0
25  ARM  0
    END
    PLO      (Produces Figure 8)
    BEG
    END
    REP
    FUR
    END
    REP
    PLO      (Produces Figure 9)
    STO
```

Table 2. Input Listing of Sample Run

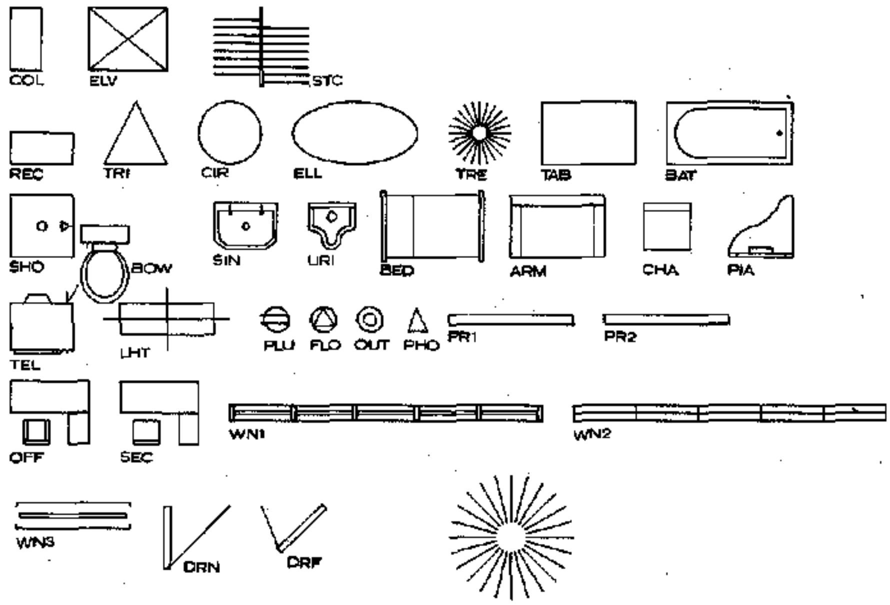


Figure 1. Graphical Vocabulary of PLAN (B. ÖZGÜÇ, "An Interactive Algorithm for Architectural Drafting", Unpublished Master of Architecture Thesis, Middle East Technical University, Ankara, 1975, p.21)

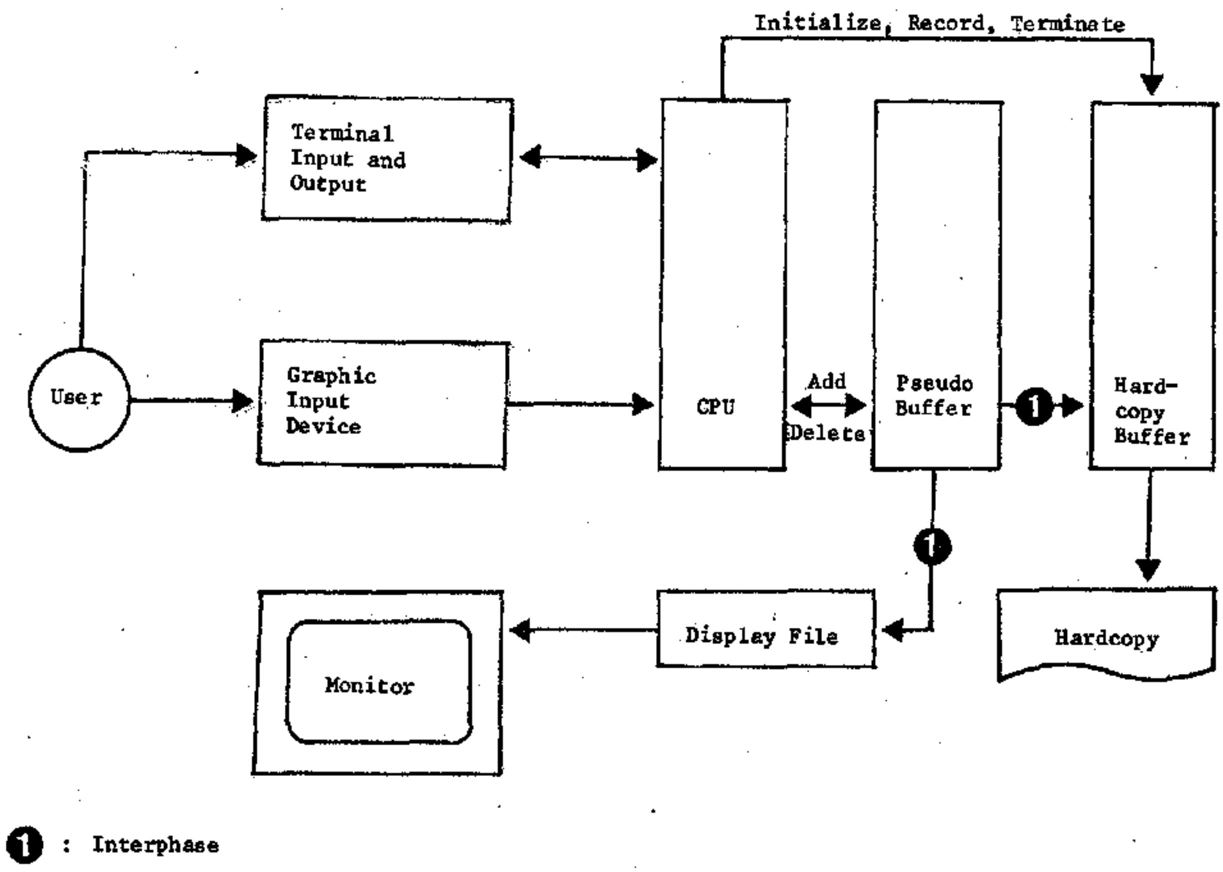


Figure 2. User-computer system utilizing PLAN

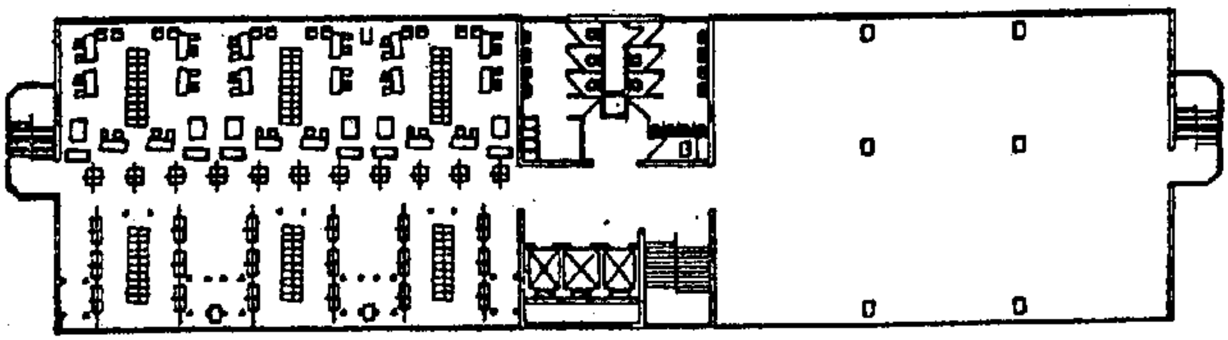


Figure 3.

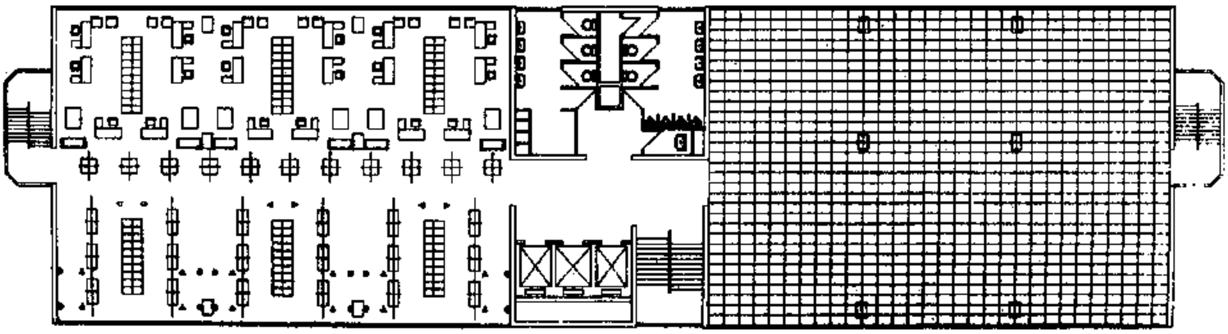


Figure 4.

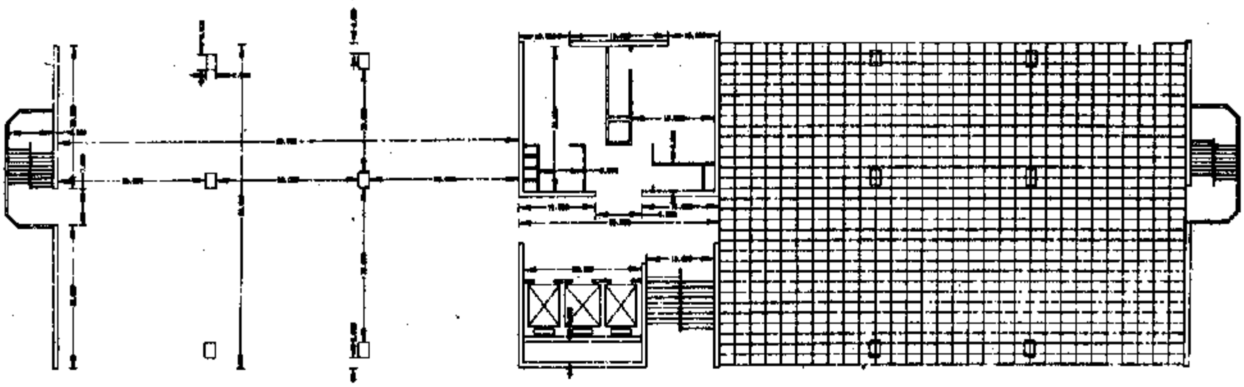


Figure 5.

9. The specification 11 COL 9 is a repetition; it re-executes all items from input specification 9 onwards. More detail can be found in the Appendix.

After entering into the running mode, fifteen items[9] are specified as given in Table 2. The succession of END and PLO commands immediately thereafter produces the structural layout shown in Figure 6. The REP and PLO commands following this produces Figure 6 again on a new page of the plotter.

The second buffer is then reached with the FUR command. Upon the specification of eight window items and the two commands END and PLO, Figure 7 is obtained. The next FUR command takes the user back into the furnishings buffer. This time seventeen items are specified and plotted. (Figure 8)

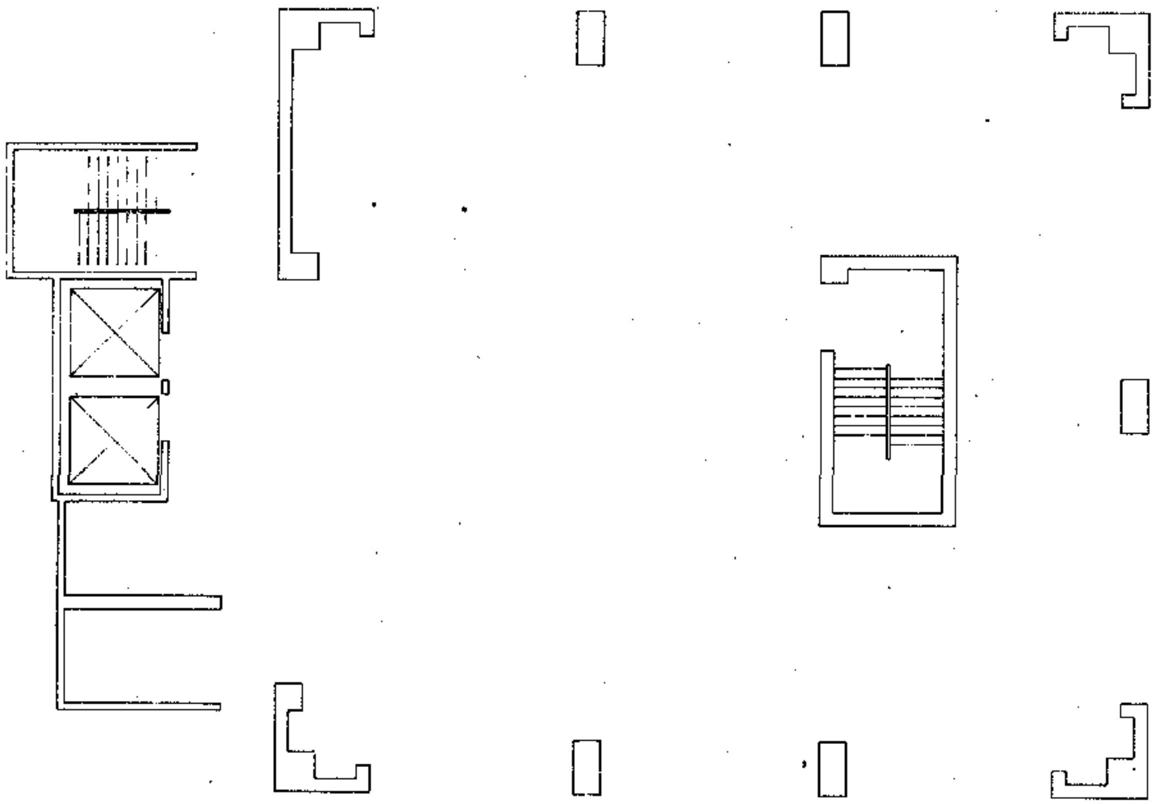
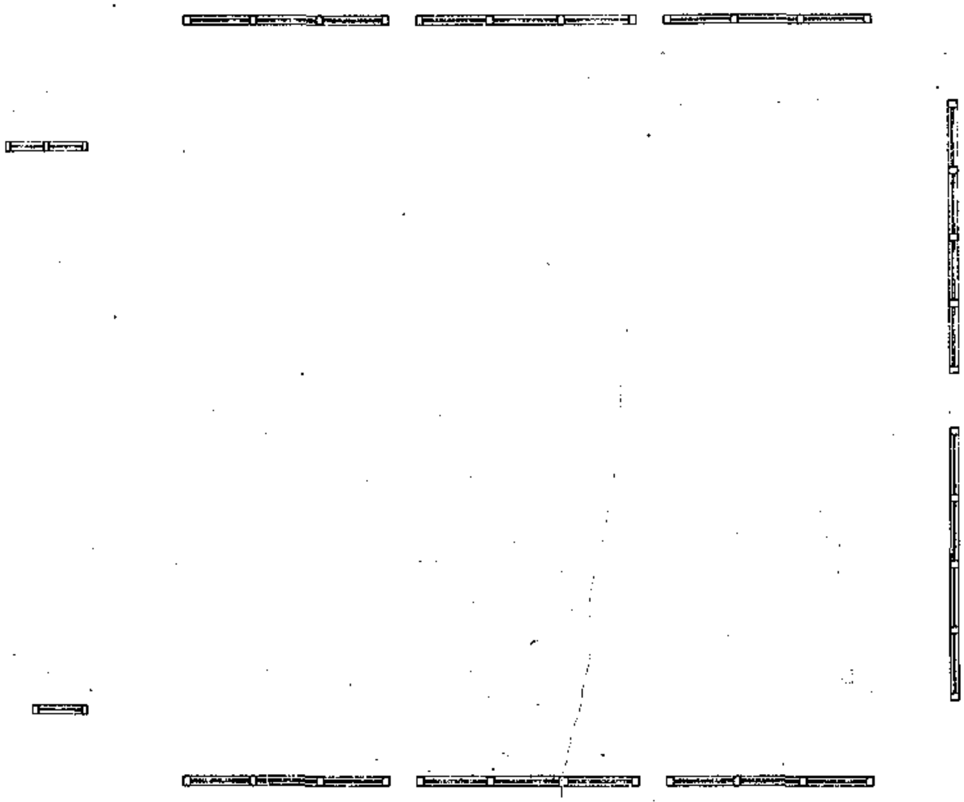


Figure 6.

Figure 7.



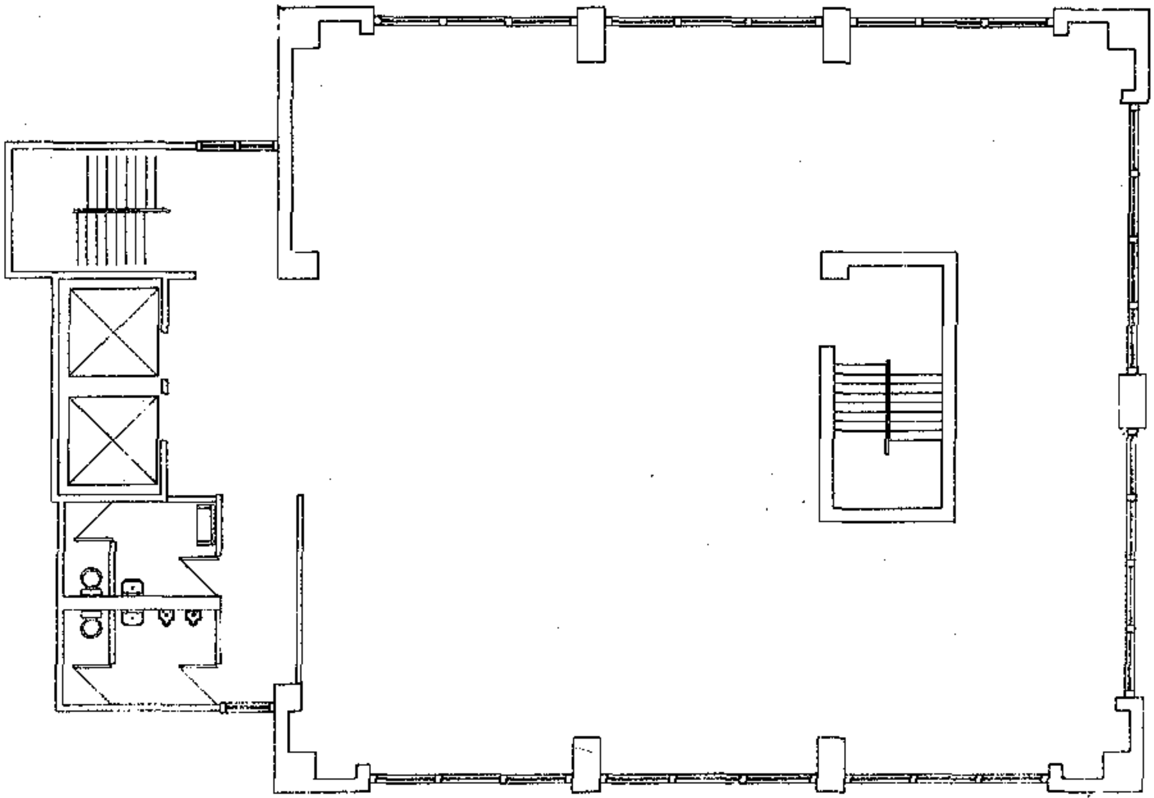
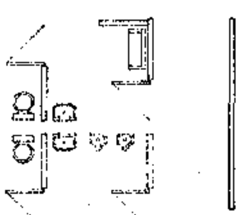


Figure 8.

Figure 9.



It is important to note here that the PLO statement by itself only plots the information that is added after the latest END command. In order to plot the whole pseudo-buffer, A REP has to be specified, which then would draw the windows together with the furnished wet core; Figure 7 would be superimposed on Figure 8.

This, however, is not the principle of overlaying that was discussed earlier. A true overlaying is shown in Figure 9 which has been obtained as follows: By first specifying BEG, the program goes into the first buffer. Then END and REP are given which re-executes the structural plan buffer without giving a hardcopy. The successive commands of FUR, END and REP re-execute the furnishings pseudo-buffer still giving no hardcopy. The final command of PLO then plots the two pseudo-buffers superimposed giving Figure 9.

Such a long process for obtaining Figure 9 is definitely not necessary. Figures 6 to 8 could directly lead to Figure 9 had there been no PLO commands in between. However, these have been given step by step in order to illustrate what can be done with the program.

APPENDIX- SOME FEATURES OF PLAN

There are five levels of programming in PLAN which should be differentiated in order to make the program general and easily adaptable to different installations. For most graphical applications, these levels are the same and even with different software, the same grouping may be achieved.

1. *Initialization and Termination*- These routines are necessary in order to set default values, clear buffers, close files, etc. Since these are partially dependent on the installation where the program is run, these have been arranged as subprograms. Two routines BEGINN and BINISH, in the present version, initialize and terminate a file on tape which is later used in driving a Calcomp plotter.
2. *Plotting*- Routines that produce the plots are considered to be another level of programming since these are also dependent on the hardware. The present version of PLAN incorporates three plotting routines PLLT, PLT and PLOTI for producing different lines and FINISH which plots everything upto that point with all the modifications and corrections and advances the Calcomp drum to a new page. Overlaying of different plans is also achieved through FINISH.
3. *Templates*- These subprograms, given in Table 1, digitize a certain form or a shape within a unit box, performing transformations on it and sending the information to the plotting routines. These are naturally another level since the templates of an architectural drafting program, for example, are not the same as those for an electronic circuit. The plotting of a typical item follows the flowchart given in Figure 10.

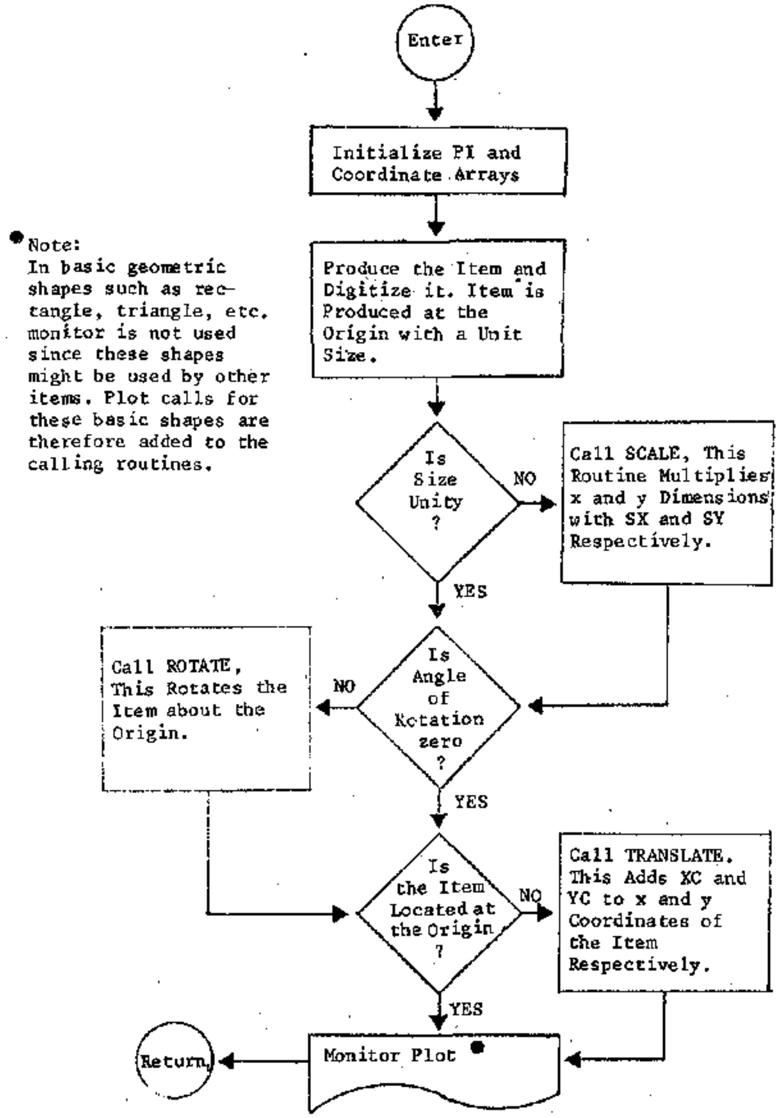


Figure 10. Plotting of a Typical Item

4. *Transformations*- These are operations on the unit pictures produced by the templates consisting of scaling (subroutine SCAL), rotation (ROTATE), and translation (TRANSL) in the present version.

5. *Controlling*- This is the main program (MAIN) where data structures are defined, user requirements are interpreted and the necessary subroutines are activated. MAIN is capable of replotting the buffers, scratching them, making additions to them when desired and deleting items together with garbage collection.

A simplified flowchart of PLAN is given in Figure 11.

The data structure of PLAN has been designed on the principle of preserving the input specification and re-executing it whenever necessary rather than storing the digitized picture. This is very effective when dealing with limited memories but when time is more generously allowed, especially so when minicomputers are used to drive graphic equipment.

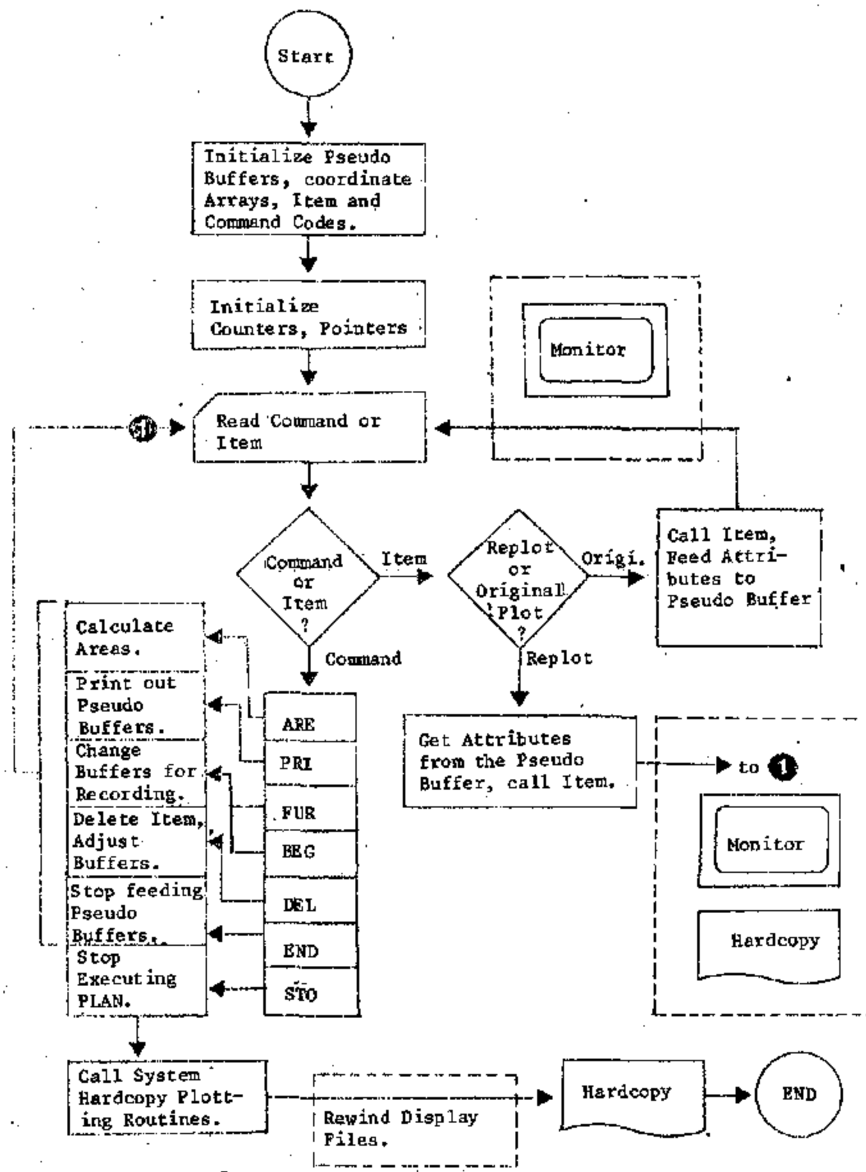


Figure 11. Simplified Flowchart for PLAN

The pseudo-buffers of PLAN are illustrated in Figure 12. If we assume this whole figure to be a pseudo-buffer, as many of these as required can be included in the program in order to produce any level of overlaid plots. Figure 12 illustrates two arrays. The first one contains the names or codes of items and two pointers for each item, shown as K_i and M_i. The second array contains the attributes of each item, such as its size, location, orientation, line intensity, etc. Pointer K_i is an integer which shows the location of the first attribute associated with the ith item. This is better explained in Figure 13. Each item has a predefined number of attributes, therefore a pointer which shows the last attribute associated with a particular item is not necessary.

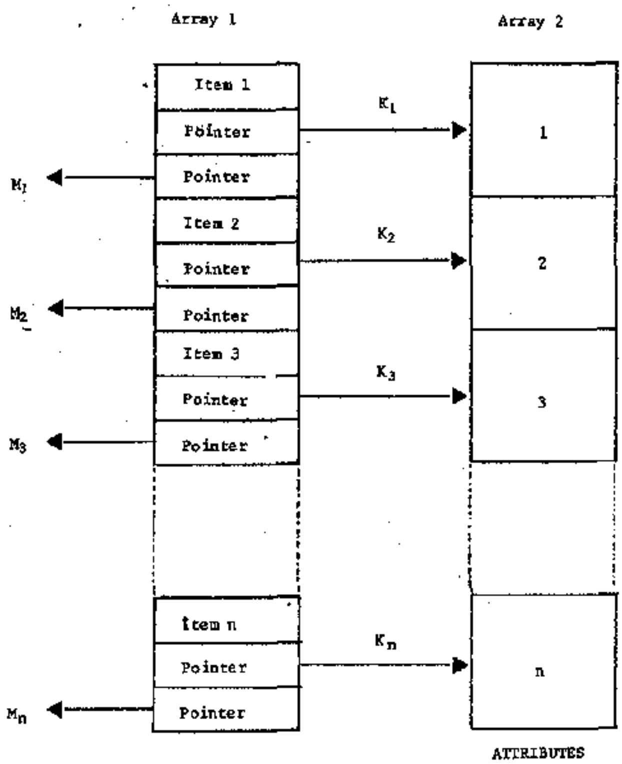


Figure 12. Data Structure of Pseudo-buffers

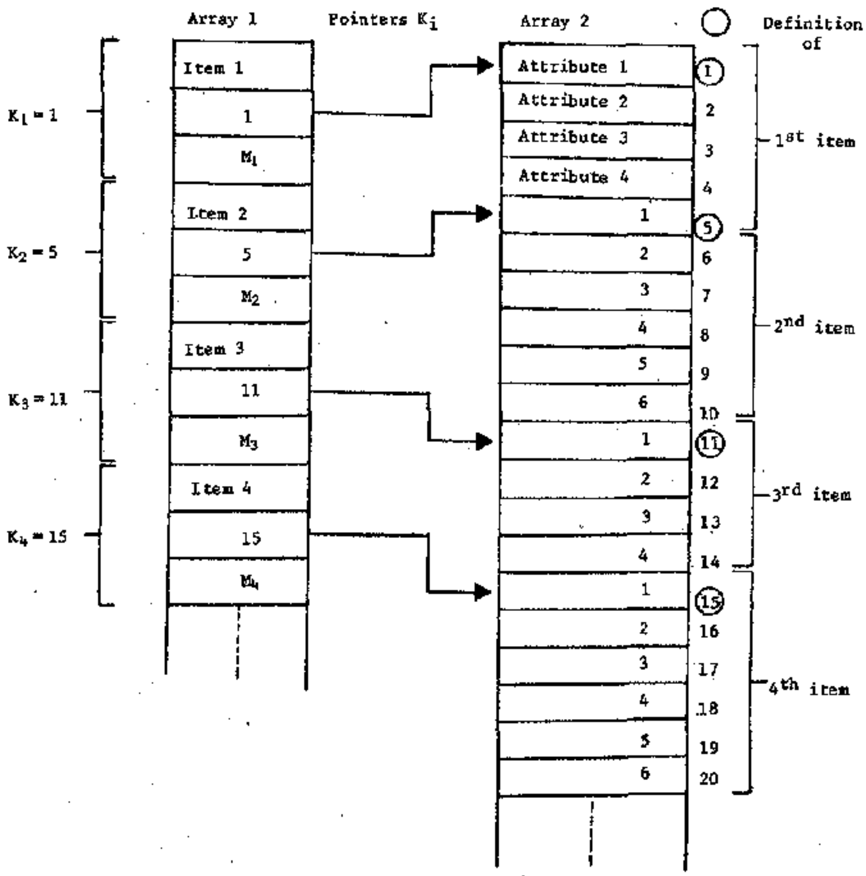


Figure 13. Functioning of Pointers K_i

○ Encircled locations are in fact the K_ith subscripts of Array 2.

When an item is deleted, the garbage collection routine makes necessary alterations and adjustments. For example, if item 3 is deleted from Figure 13, its four attributes in array 2 (locations 11 through 14 inclusive) are also automatically deleted, item 4 is brought up together with its attributes while K_4 is made 11 instead of 15. Also, the old item 4 is now sequenced as item 3. Additions are always placed at the end of the arrays.

The pointers M_i in Figures 12 and 13 are used for a different purpose. These are optional and are used only when a previously defined item is repeated within an arithmetic progression. This previously defined item is then called the base definition. The M_i pointers of the base definitions are zero since there is no use for an item to point to itself. But if an item is a repetition, then M_i is an integer pointing to the base definition for this repetition. The attributes pointed by K_i for a repetition are used to define the principles of the repetition such as the number of repetitions in the x and y directions with the number of length units between each repeated item. Figure 14 illustrates the pointers M_i. Item 4, in fact, is a dummy item and points to item 2. Item 2 is then re-executed, plotted as many times as required in a given order specified by the attributes of item 4 (subscripts 15 through 19 inclusive in array 2.) When this is finished, array 1 continues starting with item 5 which might be a base or another repetition. If

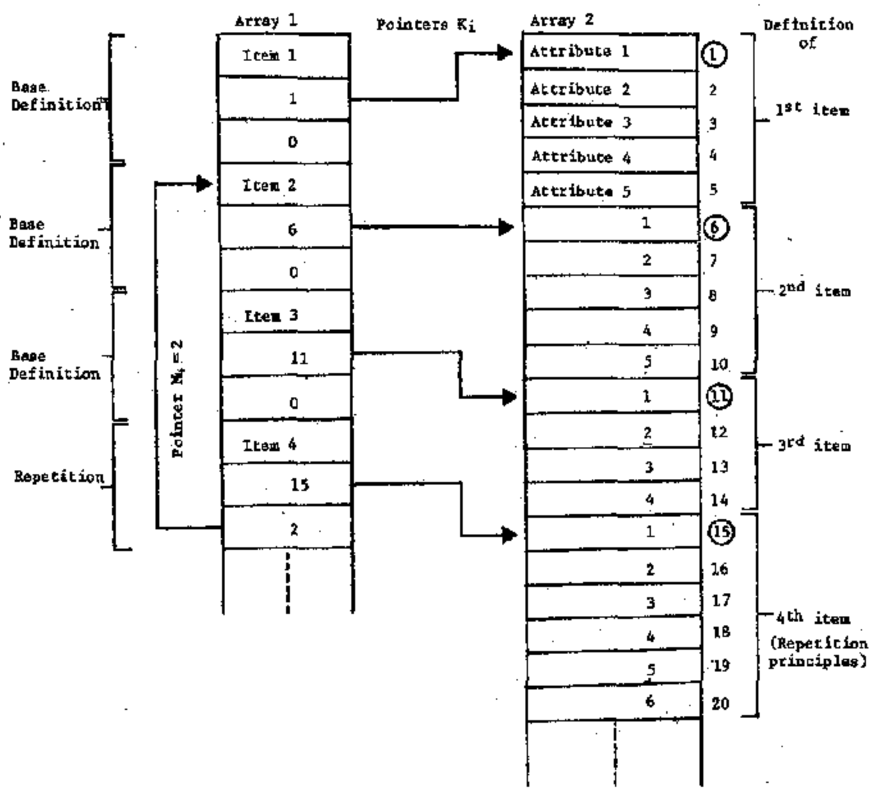


Figure 14. Functioning of Pointers M_i Pointers M_1, M_2, M_3 = 0, thus items 1, 2, 3 are base definitions.

the user wants to delete a base definition to which there are one or more M_i pointers, the program will give an error message and ask the user to delete the repetitions first.

PLAN was originally compiled with the WATFOR compiler and then adapted for FORTRAN IV G, in which it is now available.[10]

10. For more detail and a list of the program, see B. ÖZGÜÇ, "An Interactive Algorithm for Architectural Drafting", Unpublished Master of Architecture Thesis, Middle East Technical University Ankara, 1975.

MİMARİ ÇİZİMLER İÇİN BİR BİLGİSAYAR İZLENCESİ

ÖZET

Bilgisayar grafiğinin mimari tasarımı etkilendiği önemli yönlerden ikisi

a) mimari çizimlerin bilgisayarlar tarafından çizilebilmesi, ve
b) mimari tasarım sırasında seçeneklerin üretilmesi işlemine bilgisayar yeteneklerinin büyük ölçüde yardımcı olabilmesidir.

Bu yazı, her iki yönde de kullanılabilecek bir bilgisayar izlencesinin dayandığı temel noktaları tanıtmakta ve izlencenin yapısı ile kullanılışı üzerine bilgi içermektedir.

PLAN adı ile tanınan izlence bilgisayarın özek birimine ek olarak çıktı için sayısal bir çizici ile bir grafik ekrandan, girdi için ise elektronik bir tabletten oluşan bir sisteme göre tasarlanmıştır. PLAN kullanıcıya grafik sözlüğünde bulunan şekiller üzerinde boyutsal, konumsal ve yönsel dönüşümler yaparak bunlarla düzlemde bir bileşim yaratma olanağı getirmektedir. İzlence ile istenen bir çözüm elde edildiği zaman, çözümün sayısal çizicide bir düzlemsel çizimi yapılabilmektedir.

İzlencenin içerdiği grafik sözlük ve kullanım komutları bir örnek ile birlikte açıklanmaktadır.

BIBLIOGRAPHY

GALLENSON, L. *A Graphics Tablet Display for Use Under Timesharing*. Washington, D.C.: Thompson Books, 1967.

GINGER: Graphics Interaction with General Element. *ABACUS Occasional Paper*, n.65, University of Strathclyde, Glasgow, March 1975.

GREENBERG, D.P. Computer Graphics in Architecture. *Scientific American*, v.230, n.5, May 1974.

ÖZGÜÇ, B. "An Interactive Algorithm for Architectural Drafting." Unpublished Master of Architecture Thesis, Middle East Technical University, Ankara, 1975.