GRAPHIC SIMULATION WITH GPSS

M. E. Stephenson
System Simulations
The Boeing Company
Huntsville, Alabama

Boeing has developed a graphics simulation system which provides the capability of displaying simulation models and their output on an IBM 2250 display unit. The system is based on BMT (Boeing Modeling Technique), which is an expanded version of GPSS II for use with the IBM 360 Operating System (OS). Existing GPSS II models may be executed under BMT since it has GPSS II as a subset.

At Boeing, the need for graphics grew out of a need for faster check out of new models. For example, Boeing's Saturn V prelaunch model must be reprogrammed each time the countdown sequence changes. Two weeks of coding and three weeks of check out are required to put a new model into production. Using graphics to debug on line, the check out time was reduced to one week. Two runs, taking five or six days to process in a batch environment, were needed to eliminate keypunch and syntax errors alone. This can be done in an hour at the 2250. And, overall shop through put is not seriously hampered by the graphics operation since batch processing can run concurrently with the graphics in other partitions.

A BMT model (without graphics) is processed in four distinct phases, each phase by a separate module. These four modules operate in an overlay structure controlled by a fifth module. In addition to the control program, the fifth module has all of the I/O routines used by the other four. The four phases of processing are; assembler, input, execute, and output. The assembler module assigns absolute values for all symbolic names used in the source model and detects certain syntax errors in the source statements. The assembler outputs a listing of the modeler's source statements (assembly listing) with diagnostics of errors detected in the first pass and a symbol table. Input builds the tables in core storage in which the execution time statistics will be complied. Input produces a listing of the model (the input listing) with the assigned values for each symbol and any additional diagnostics for statements flagged either in pass two of assembler or in input itself. The execution module performs the actual simulation - compiling statistics in the tables set up in core by input. Depending on the logic of the model, some information may be output during the execution phase of the model. Most of the results of the simulation are maintained in the tables in core to be compiled and printed as reports by the output module. Figure 1 shows a diagram of the BMT system without graphics.

When graphics is added to the BMT system, the four types of output (which were printed previously) are reduced to three types by combining the output from the execution and output phases into one data set. These three types of output are written onto disk and form the three basic types of display. The assembler and input listing displays are shown in figures 6 and 7. The four basic modules of BMT remain unchanged in the graphics version, but the control program module is replaced completely. Figure 2 shows the BMT system with graphics.

The graphics simulation model may be introduced into the 360 Operating System for processing via the normal input stream or from the 2250 by use of the Terminal Job Schedular (TJS). In either case the necessary Job Control Language (JCL) statements are invoked from the OS procedure library. Figure 3 shows the procedure as displayed by the TJS. A light pen detect on the word "SCHEDULE" of this panel causes the TJS to enter the job into the OS job queue for processing as though it had been entered through the normal input stream.

When the BMT graphics job is scheduled, it assembles the first model and displays the option panel shown in Figure 4. This panel is also displayed after the input and output phases. As indicated by the panel, the user may elect to continue processing or display his output by selecting the proper option with the light pen. If the user chooses to continue processing, the next phase of processing would be entered (e.g. input if he had just finished assembling etc.).

If the option to display is selected, the panel shown in Figure 5 is

displayed. It gives instructions for updating a page of the assembler listing. An end key attention while this panel is being displayed brings up the first page of the assembler listing as shown in Figure 6. Although the first page of the assembler listing is always displayed first, all of the output accumulated is available. Therefore, if input has finished processing, its listing may be displayed and if output has finished processing, all three types of output may be displayed. The next or previous page of output is selected with the light pen on those options in the footer line of the display. Any page may be selected by entering its number from the alphanumeric keyboard and selecting the option "PAGE" with the light pen. A word in the footer line indicates the last phase of processing completed.

The assembly listing may be updated from the console using the light pen, programmed function keys, and the alphanumeric keyboard. The graphics program is placed in the update mode by depressing function key 5. To change a line of the assembler listing the line is selected with the light pen, which places a cursor at the beginning of that line. Using the alphanumeric keyboard, the line may be modified in anyway the user wishes. Depressing the end key signifies completion of the line. To insert a line, line 20 is selected with the light pen and the number where the line is to be inserted followed by the letter I (e.g. 16I) is keyed in. Similarly, a line may be deleted by selecting line 20 and entering the line number followed by the letter D (e.g. 11D). After all of the changes on a particular page have been made, function key 4 must be depressed to record the changes on disk and take the program out of update mode. An attempt to select another page when in update mode will cause an error message to be displayed.

To continue into the next phase of processing after seeing the output displayed, function key 6 is depressed. If updates to the assembler listing have been made the program will automatically reassemble the updated model. If the model has finished processing and no updates were made, the next model is assembled. Control will return to OS after the last model or the user can terminate processing at anytime by depressing the cancel key.

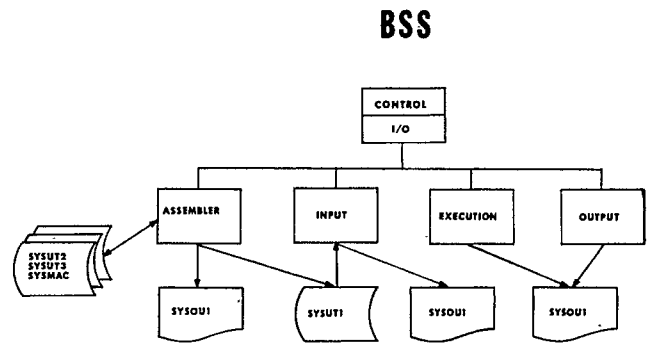

Fig. 1

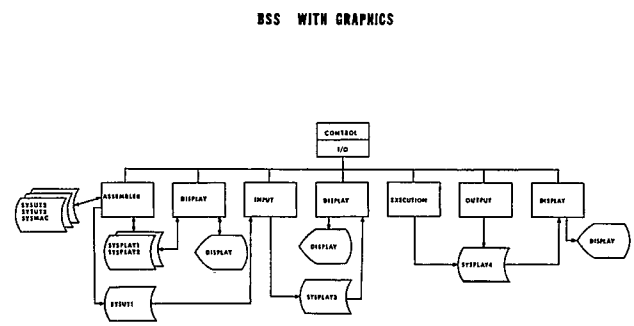

Fig. 2

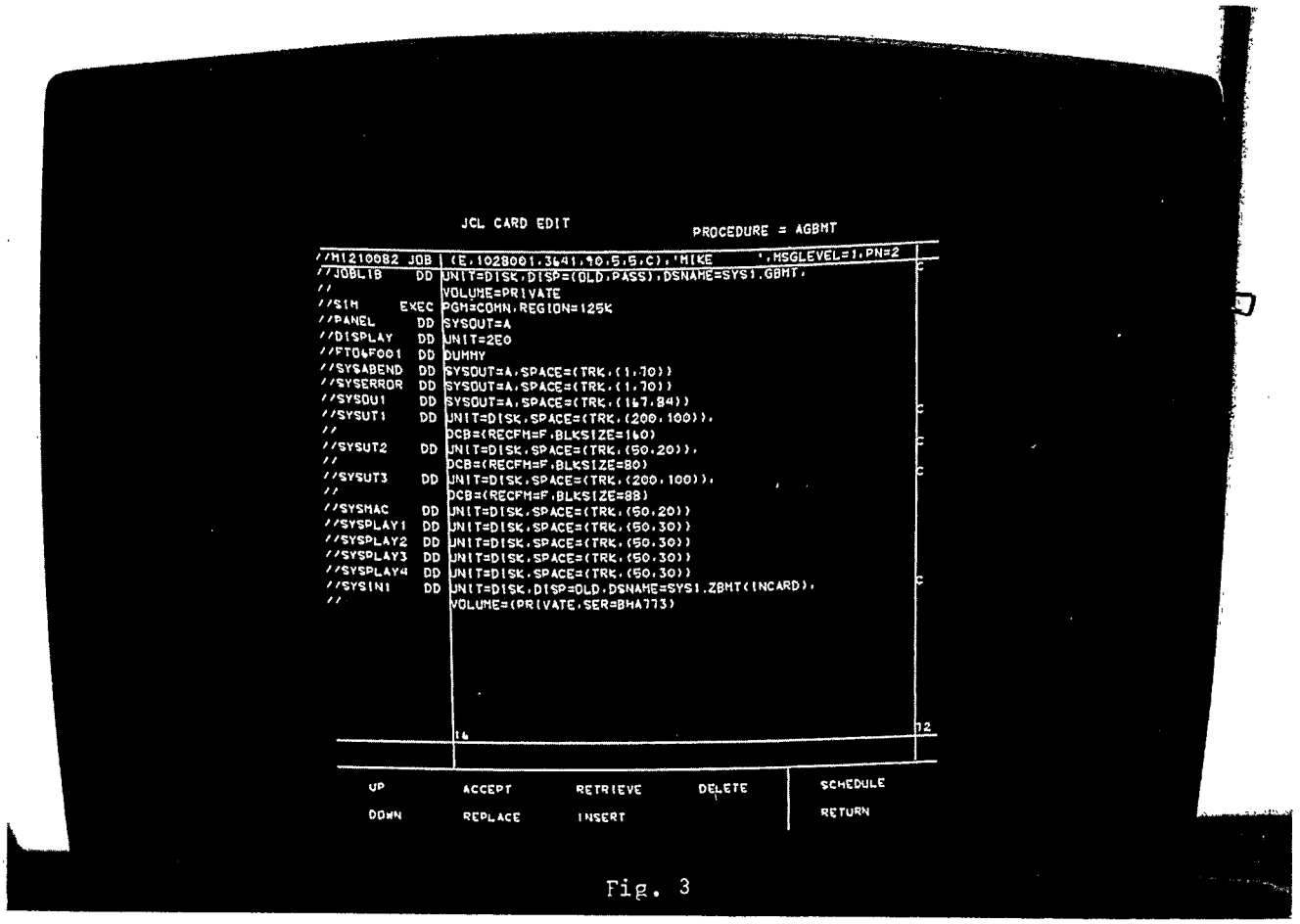
JCL CARD EDIT
PROCEDURE = AGBMT
//H1210082 JOB (E,1028001,3641,10,5,5,C),'MIKE ',MSGLEVEL=1,PN=2
//JOBLIB DD UNIT=DISK,DISP=(OLD,PASS),DSNAME=SYS1.GBMT,
// VOLUME=PRIVATE
//SIM EXEC PGM=COMN,REGION=125K
//PANEL DD SYSOUT=A
//DISPLAY DD UNIT=2E0
//FT06F001 DD DUMMY
//SYSABEND DD SYSOUT=A,SPACE=(TRK,(1,70))
//SYSERROR DD SYSOUT=A,SPACE=(TRK,(1,70))
//SYSOU1 DD SYSOUT=A,SPACE=(TRK,(167,84))
//SYSUT1 DD UNIT=DISK,SPACE=(TRK,(200,100)),
// DCB=(RECFM=F,BLKSIZE=160)
//SYSUT2 DD UNIT=DISK,SPACE=(TRK,(50,20)),
// DCB=(RECFM=F,BLKSIZE=80)
//SYSUT3 DD UNIT=DISK,SPACE=(TRK,(200,100)),
// DCB=(RECFM=F,BLKSIZE=88)
//SYSMAC DD UNIT=DISK,SPACE=(TRK,(50,20))
//SYSPLAY1 DD UNIT=DISK,SPACE=(TRK,(50,30))
//SYSPLAY2 DD UNIT=DISK,SPACE=(TRK,(50,30))
//SYSPLAY3 DD UNIT=DISK,SPACE=(TRK,(50,30))
//SYSPLAY4 DD UNIT=DISK,SPACE=(TRK,(50,30))
//SYSIN1 DD UNIT=DISK,DISP=OLD,DSNAME=SYS1.ZBMT(INCARD),
// VOLUME=(PRIVATE,SER=BHA773)
14
72
UP
ACCEPT
RETRIEVE
DELETE
SCHEDULE
DOWN
REPLACE
INSERT
RETURN

Fig. 3

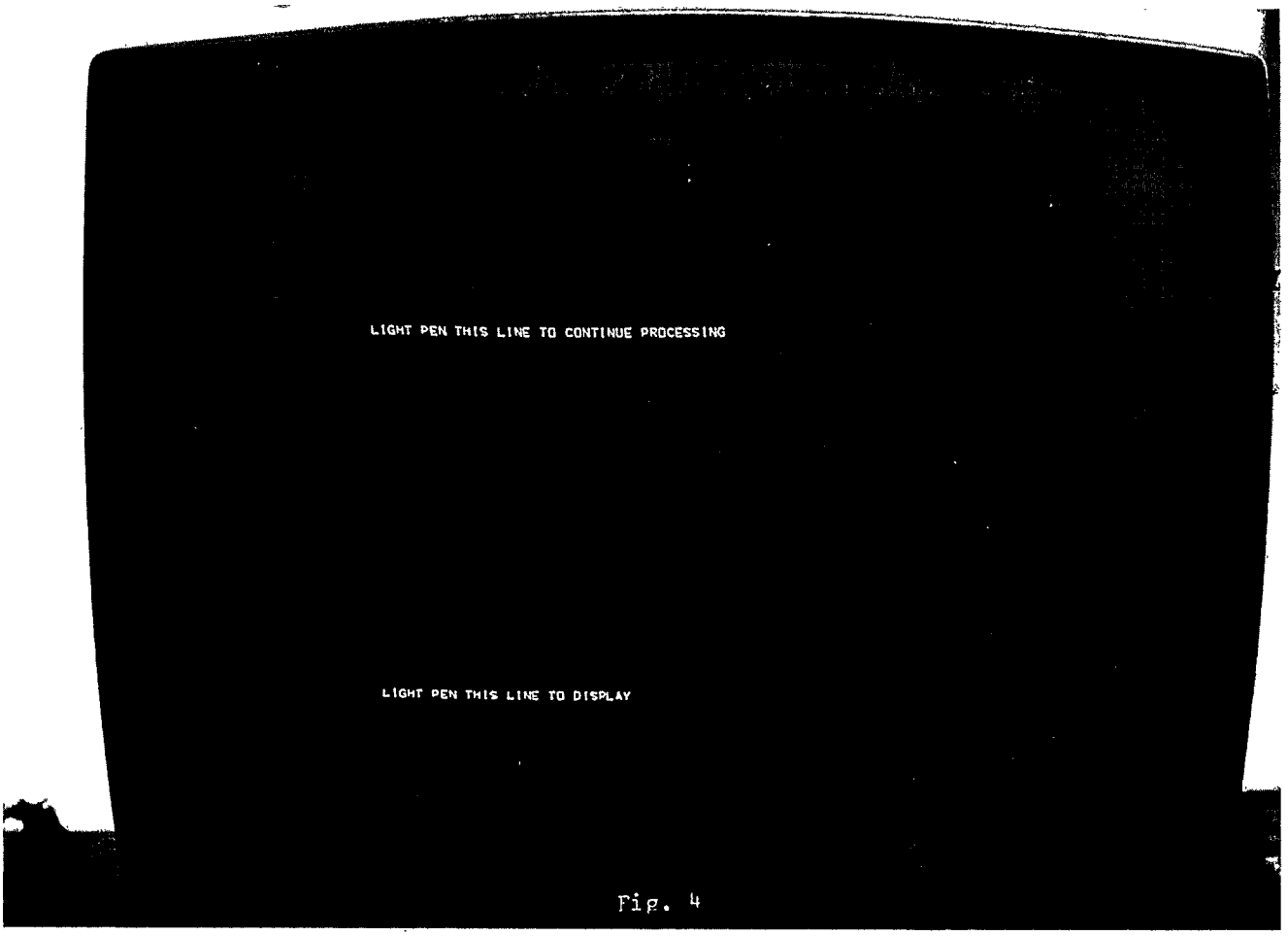
LIGHT PEN THIS LINE TO CONTINUE PROCESSING
LIGHT PEN THIS LINE TO DISPLAY

Fig. 4

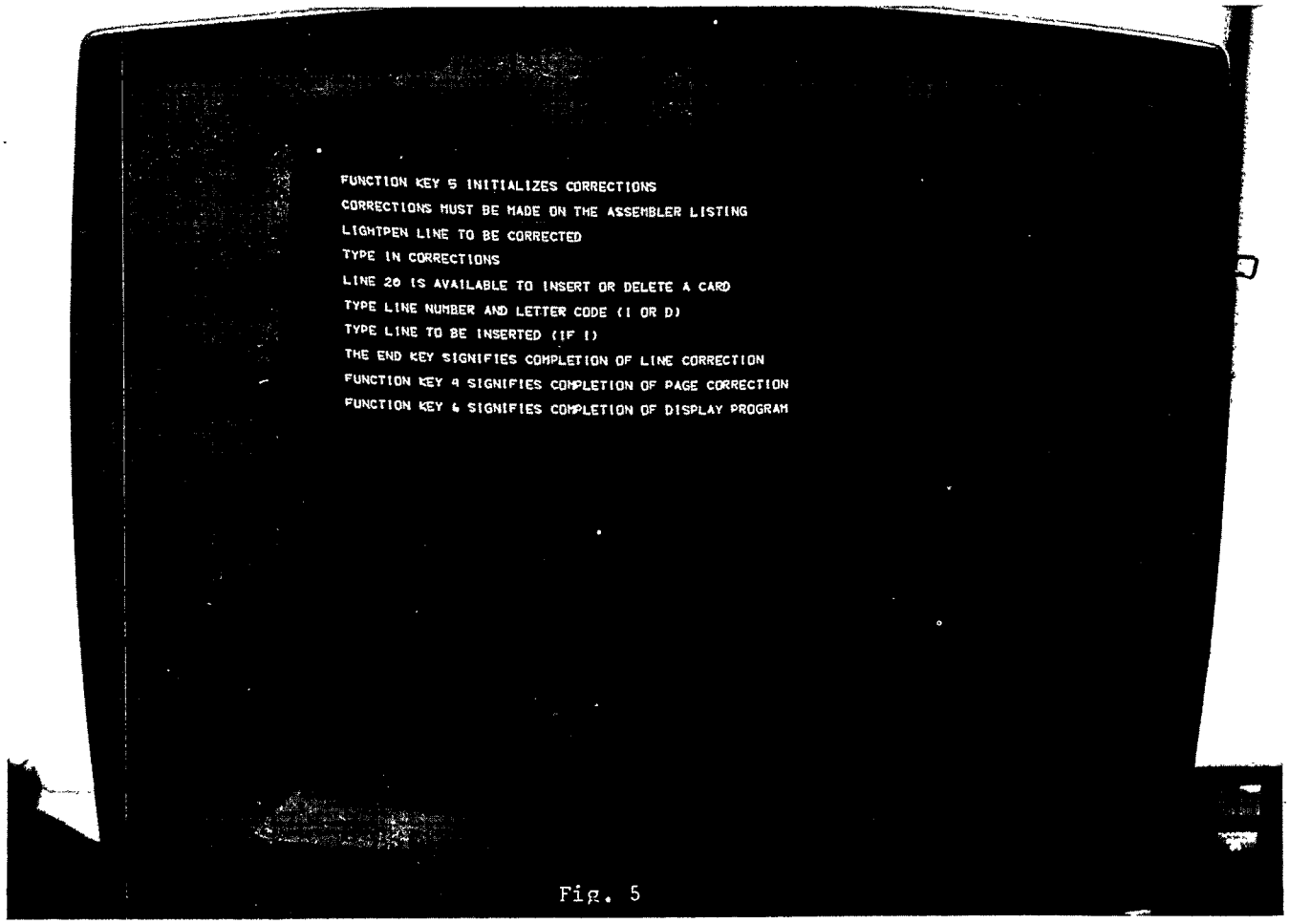
FUNCTION KEY 5 INITIALIZES CORRECTIONS
CORRECTIONS MUST BE MADE ON THE ASSEMBLER LISTING
LIGHTPEN LINE TO BE CORRECTED
TYPE IN CORRECTIONS
LINE 20 IS AVAILABLE TO INSERT OR DELETE A CARD
TYPE LINE NUMBER AND LETTER CODE (I OR D)
TYPE LINE TO BE INSERTED (IF I)
THE END KEY SIGNIFIES COMPLETION OF LINE CORRECTION
FUNCTION KEY 4 SIGNIFIES COMPLETION OF PAGE CORRECTION
FUNCTION KEY 6 SIGNIFIES COMPLETION OF DISPLAY PROGRAM

Fig. 5

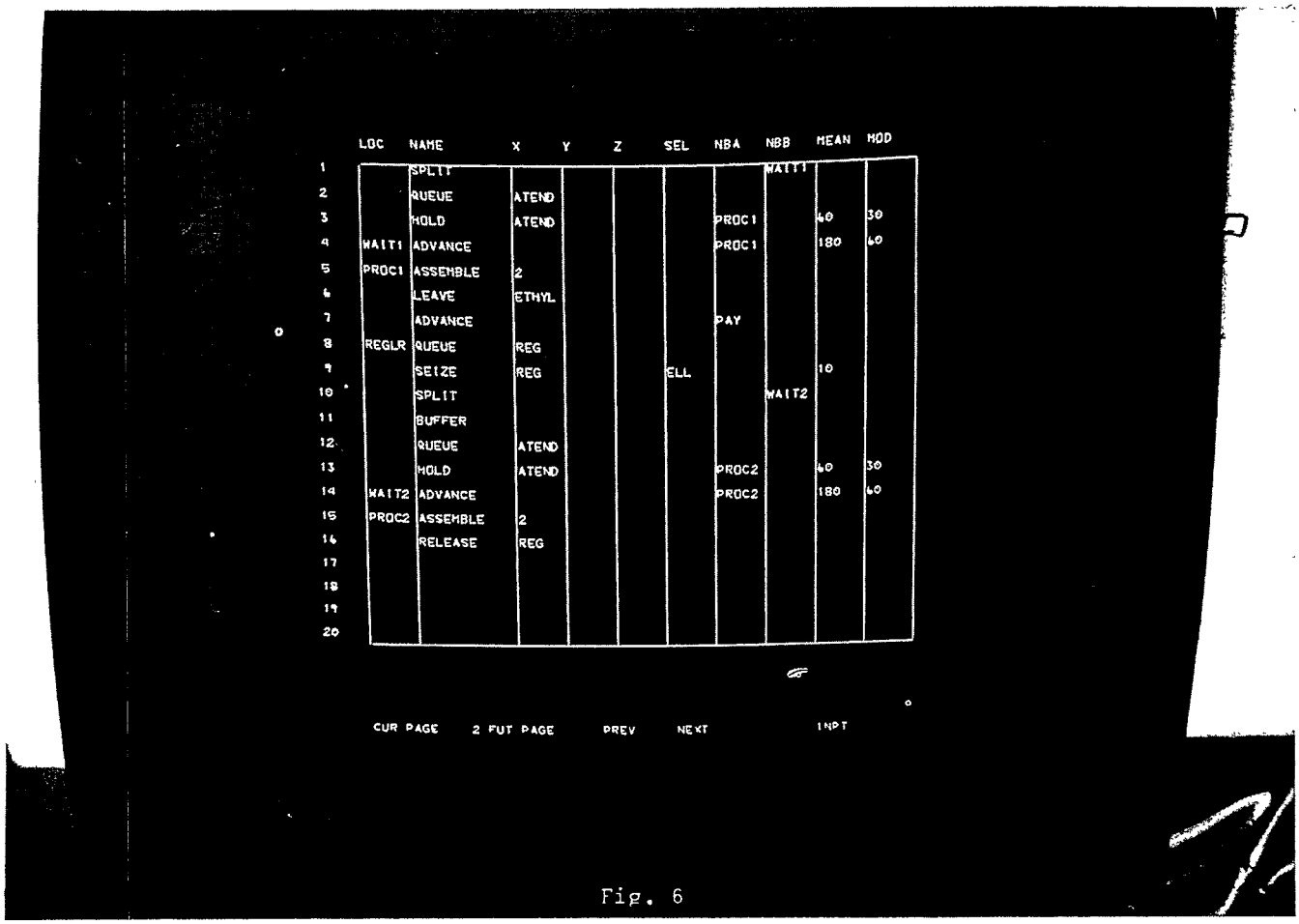
LOC NAME X Y Z SEL NBA NBB MEAN MOD
1 SPLIT WAIT1
2 QUEUE ATEND
3 HOLD ATEND PROC1 40 30
4 WAIT1 ADVANCE PROC1 180 60
5 PROC1 ASSEMBLE 2
6 LEAVE ETHYL
7 ADVANCE PAY
8 REGLR QUEUE REG
9 SEIZE REG ELL 10
10 SPLIT WAIT2
11 BUFFER
12 QUEUE ATEND
13 HOLD ATEND PROC2 40 30
14 WAIT2 ADVANCE PROC2 180 60
15 PROC2 ASSEMBLE 2
16 RELEASE REG
17
18
19
20
CUR PAGE 2 PUT PAGE PREV NEXT INPT

Fig. 6

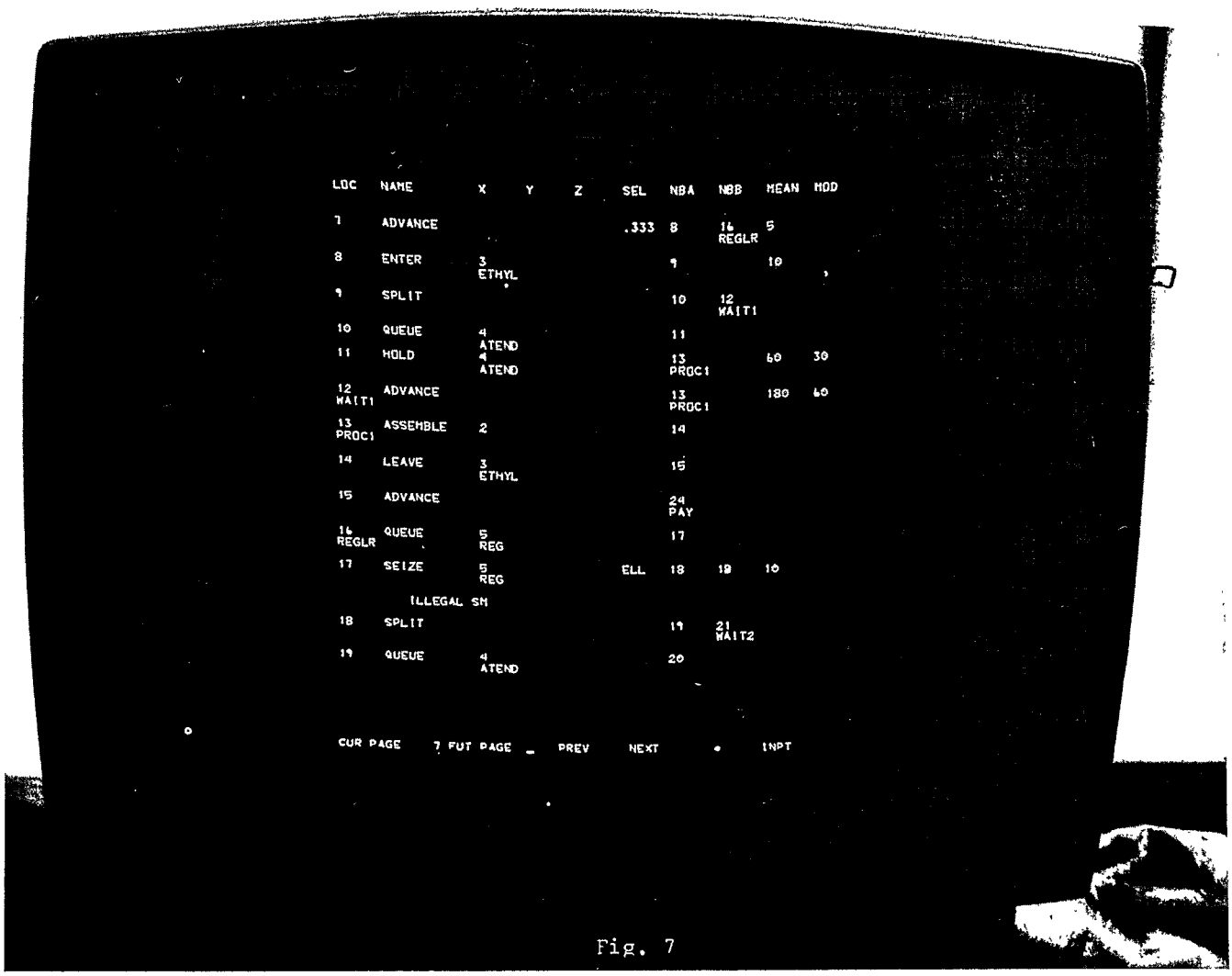
LOC NAME X Y Z SEL NBA NBB MEAN MOD
1 ADVANCE .333 8 16 REGLR 5
8 ENTER 3 ETHYL 9 10
9 SPLIT 10 12 WAIT1
10 QUEUE 4 ATEND 11
11 HOLD 4 ATEND 13 PROC1 60 30
12 WAIT1 ADVANCE 13 PROC1 180 60
13 PROC1 ASSEMBLE 2 14
14 LEAVE 3 ETHYL 15
15 ADVANCE 24 PAY
16 REGLR QUEUE 5 REG 17
17 SEIZE 5 REG ELL 18 19 10
ILLEGAL SM
18 SPLIT 19 21 WAIT2
19 QUEUE 4 ATEND 20
CUR PAGE 7 FUT PAGE _ PREV NEXT • INPT

Fig. 7