THE ICGS SYSTEM: USERS MANUAL

by

David Henry Mueller

MASTER

February 1974

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN · URBANA, ILLINOIS

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

-UIUCDCS-R-74-608

THE ICGS SYSTEM: USERS MANUAL

by

David Henry Mueller

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the United States Atomic Energy Commission, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

February 1974

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
URBANA, ILLINOIS 61801

* Supported in part by contrac US AEC AT(11-1)2383

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

See

Table of Contents

1.	Introduction.....	1
2.	System Structure.....	2
3.	Data Structures of the System.....	7
4.	Description of the Initializer Tables.....	10
Appendix A The Production Initializer.....		14
Appendix B An Example of the Table Module.....		21
Bibliography.....		26

ICGS System

System Tables

1. Introduction

The Illinois Computing Graphics System (ICGS) is a batch system for a mini-computer. It executes a limited number of system programs, and provides facilities for rapid overlaying of these programs.

The system was written for the following configuration:

- 1) PDP-11/20 with 16K of core memory
- 2) KE-11A Extended Arithmetic Element
- 3) Teletype
- 4) RF11 256K-2M word fixed-head disk
- 5) 2 TC11 Dectapes
- 6) GDI model 100 card reader
- 7) Gould 4800 printer/plotter

The System is intended to run under DOS. The System itself consists of a core resident Supervisor, a set of disk resident System modules, a set of desk resident user-defined modules, two bootstrap loaders, and an independent program to aid in the building of user modules. This Manual is intended to aid the System programmer in setting up system tables. Familiarity with the PDP-11 and DOS is assumed.

2. SYSTEM STRUCTURE

The entire system runs under control of and with the services of the Disk Operating System (DOS). From the point of view of DOS, the system looks like a single program, a portion of which is permanently core resident and the remainder of which resides on storage devices in the form of overlays. The core resident portion is the Supervisor and is responsible for providing the overlaying facilities. DOS, then, views the system as in Figure 2.1.

The Supervisor views the remainder of the system as a set of monitors, each of which may or may not possess a set of overlays. Each of the monitors and overlays is identified to the Supervisor by an ordered pair of numbers. In the case of an overlay, the pair is (monitor number, overlay number). In the case of a monitor, the pair is (0, monitor number), since the Supervisor is monitor 0. The sets of overlays are structurally mutually exclusive, but the structural information for two overlays contained in the Supervisor may, in fact, refer to the same overlay file.

The monitors are responsible for notifying the Supervisor of changes in the status of the overlays, calling the Supervisor to read in overlays, and directing the Supervisor what to do when the monitor is done. Options for a monitor upon completion include running another monitor, turning to the batch input device for the next job, or returning to a previous monitor. The latter option allows a monitor to temporarily

Figure 2.1
System as viewed by DOS

Figure 2.2
System as viewed by the Supervisor

relinquish control to another monitor, then regain control later.

It is the responsibility of the monitor to keep track of which overlays are in core and at what points another overlay is required. Since the Supervisor cannot monitor I/O operations, the monitor itself must also notify the Supervisor when the location of an overlay file has changed. A monitor knows the location of its file by the file name; the Supervisor, upon request, looks this file up in the directory of one of the mass storage devices and records the physical location of the file in its tables. The Supervisor can then use this information to read the file in rapidly when requested to do so.

All the facilities of DOS are also available to a monitor. In effect, the Supervisor and DOS combined appear to a monitor as a single operating system. Further, a monitor knows only of its own overlays; it may not have access to another monitor's overlays. Thus, to a monitor the system appears as in Figure 2.3.

For purposes of batch execution, the monitors are grouped into sequences known as subsystems. The Grafix subsystem, for instance, consists of the Grafix compiler monitor and the Basic Plotting Package monitor. The structure of these subsystems is not contained in the Supervisor or its tables--it is defined by the monitors as each requests the next in the subsystem. The Supervisor maintains the identity of the first monitor in each subsystem and the subsystem name by which the user communicates with the system.

The system itself consists of five parts: the core-resident Supervisor, the supporting system monitors, the user-supplied monitors and overlays, the table initializer, and the debugging support. The

Figure 2.3
System as viewed by a monitor

Supervisor consists of the Sequencer, the Overlayer, and the Table Manager. There are two system monitors--one for using the Gould as a line printer, and one for handling system errors. The Initializer is a bootstrap routine which sets up the system tables and locates the monitors and overlays on the storage devices. Finally, there is a program to build overlay files and a version of the Initializer specifically designed for debugging.

3. DATA STRUCTURES FOR THE SYSTEM

The Supervisor routines are essentially table driven. The tables were set up to allow maximum flexibility in the structure and interrelationships of the monitors and overlays. The tree-like structure of the lines of control (see Figure 2.2) is the only structural constraint imposed; the remainder of the structure is defined by the tables.

The information contained in the tables includes, for all overlays and monitors, the location and length of the overlay on the mass storage devices and the starting address for the overlay in core. In addition, monitors require tables of their transfer addresses, the number of overlays per monitor, and the region of core occupied by each monitor and its overlays. The latter is required in order to initialize the stack pointer before a monitor is read in. This gives as much stack space as possible to each monitor while preventing a monitor from overwriting the stack with an overlay. Subsystem structure is defined by a table giving subsystem names and a corresponding table giving the first monitor in each subsystem.

All tables except the overlay table are simply vectors containing the information. The overlay table is constructed in a hierarchical manner to provide as much protection for the overlays as possible (see Figure 3.1). Each monitor is allocated a subtable containing the information for its overlays. In addition, the Supervisor, in its role as a monitor, is allocated a subtable containing the information for the monitors. The

Supervisor remembers the identity of the currently executing monitor and allows it--using the overlay routine--to access only its own overlays. This provides some protection for the table and corresponding overlays.

The size of each subtable is determined at load time when the initializer is run. The subtables are allocated core space at that time and may be initialized then as well. The information in the subtables may also be changed, however, using the table managing routine. This would allow, for example, a monitor to move an overlay from tape to disk, then ask the Supervisor to change its tables to reflect the new situation. The length of the overlay and its starting address in core may also be changed. As is the case with the overlay routine, the table manager allows a monitor to access only the information on its own overlays.

Initialization of the tables is accomplished by a routine executed as a bootstrap to the system. The routine exists in two versions, differing only in the source of the data to be used in the initialization. In the version to be used in production runs, the data is contained in the routine itself and is presumed to be in the correct form; little checking is done for consistency. The version used for debugging uses data obtained from an input file such as a card reader or a disk file. This data is not presumed to be correct and extensive format and consistency checking is done.

The reasoning behind the use of two versions is as follows: In the case of production runs, the bootstrap should be fast, reliable, and easy to run, so that an operator with a minimum amount of training can

execute the system with the greatest ease. Flexibility is not a great consideration in this case since the structure of the system as a whole will not change between production runs.

However, in the case of debugging runs where new portions of monitors are being tested, the situation is exactly reversed. The operator is an experienced programmer, and the speed of the bootstrap is not as important as its flexibility. Since the structure of the system, particularly the size, location, and starting addresses of monitors and overlays, will almost certainly change between runs, the operator must be able to easily alter the initial data put in the Supervisor tables. This is best accomplished if the data is taken from an input file rather than stored within the Supervisor's load module.

The remainder of this report describes the table setup for the production version of the Initializer. The input file format for the debugging version is described in the users manual.

4. Description of the Initializer Tables

The production version of the Initializer has the responsibility of allocating space in the monitor table and initializing values in all the Supervisor Tables. The Initializer draws its information from a set of tables which is in an assembly module which is linked to the Initializer and loaded with it at execution time.

There are two sets of tables for the Initializer. The first set describes the monitors and their overlays. The Initializer uses this information to allocate space for the monitor and overlay descriptions in the Supervisor's monitor table. The table entries can be filled in at this time, or they can be left blank and be filled in later by the Table Manager.

The second set of tables describes the subsystems. Values in these tables are simply entered in the appropriate Supervisor tables, since there is no allocation taking place.

The second set of tables describes the subsystems. Values in these tables are simply entered in the appropriate Supervisor tables, since there is no allocation of core space taking place. All tables have -1 as the default value. Thus, if an entry is not initialized, it is -1.

4.1 The Monitor Tables

The tables for the monitors and overlays include vectors containing the number of overlays for each monitor (INTABL), the starting addresses of each monitor (ISTART) and the largest space that the monitor and its overlays will occupy in core (REGION). INTABL is a vector of bytes, one for each monitor, containing the number of overlays for the corresponding monitor. The Supervisor, by convention, is monitor 0, and monitor 1 is a dummy. Thus, the first entry in the vector is one more than the number of monitors, while the second entry is -1.

ISTART is a vector of words, one for each monitor, giving the starting address of the monitor. This is the address to which the Scheduler branches to execute the monitor. The first entry is by convention for the line printer emulator, and is 072632_8 , while the second is for the error routine, and is 073412_8 .

REGION gives the size, in bytes, of the largest area of core ever occupied at one time by the monitor and its overlays. This is used by the Scheduler to set the stack pointer before executing the monitor so that the stack is not overwritten by an overlay.

The largest of the monitor tables is the overlay table. This is the table used to fill in the entries in the Supervisor Tables. There is one entry in the table for each possible monitor and overlay number. All monitors are listed first, followed by the overlays in the order of their monitors.

Each entry consists of two required parameters and up to five optional ones. The first two parameters are required, and are one byte each. Both are equal to zero for monitors and overlays which are not to be initialized.

- 1) Number of bytes to follow, i.e. twice the number of optional parameters present.
- 2) Status byte, set up as in Figure 4.1.

The Status Byte

Figure 4.1

The remaining parameters are optional, and are present as required by the Status byte. All are fullword parameters and must be in the order given, if they are present.

- 1) Length of the monitor as overlay, in words
- 2) Memory address where the monitor or overlay is to be loaded into core
- 3) Extension of the file name where the monitor or overlay is located, in Radix 50
- 4) Final three characters of the file name of the file where the monitor or overlay is located, in Radix 50
- 5) Initial three characters of the file name of the file where the monitor or overlay is located, in Radix 50

4.2 The Subsystem Tables

The Subsystem tables consist of two vectors and a parameter. The parameter, NSSYS, is one byte in length and gives the number of subsystems present in the system. The vector ISYSNM contains one word for each subsystem and gives the first three characters of the subsystem name in Radix 50. The subsystem name is read from the card reader and is used by the Scheduler to identify the subsystem. The remaining vector, ISYSTB, contains one byte for each subsystem, and gives the number of the first monitor in the subsystem.

4.3

The tables should all be assembled in one module. All tables should be global, but they may be in any order. An example is given in appendix B.

APPENDIX A

THE PRODUCTION INITIALIZER

INIT ; MACRO V004A PAGE 1

```

1
2      ; PROGRAM NAME      SYSTEM INITILIZOR
3
4      ; AUTHOR/DATE      DAVID MUELLER      SEPTEMBER, 1972
5
6      ; MACHINE/LANGUAGE  PDP-11/20, 16K      PAL-11R V005A
7
8      ; PURPOSE          TO ALLOCATE AND INITILIZE SYSTEM TABLES. THIS
9                          PROGRAM IS MEANT TO BE USED FOR PRODUCTION RUNS
10                         UNDER THE IGCS- SYSTEM.
11
12      ; SPECIFICATION STATEMENTS
13
14      ; .TITLE INIT
15      ; .GLOBL INTABL, ISTART, NSSYS, ISYSNM, ISYSTB, SYSEND, INITBL, REGION
16      ; .GLOBL NTABLE, START, NSYS, SYSNAM, SYSTAB
17      ; .GLOBL SYSMOD, CORTOP, MODTBL, MODEND, SVC, UPDATE, FILERR, PRLINK
18      ; .CSECT
19      000000      R0=R0
20      000000      R1=R1
21      000001      R2=R2
22      000002      R3=R3
23      000003      R4=R4
24      000004      R5=R5
25      000005      SP=R6
26      000006      PC=R7
27
28      ; DIRECT ASSIGNMENTS
29
30      ; CR=15
31      000015      CR=15
32      000012      LF=12
33
34      ; INITILIZE I/O AND VARIABLES
35
36      000000 012746 INIT:  MOV  #PRLINK, -(SP)      ; INIT TTY
37      000004 000376      EMT  6
38      000006 016767      MOV  MODTBL, AMODTB      ; INITILIZE BASE ADDRESS INDICATOR
39      000014 012700      MOV  #INITBL, R0          ; SET POINTER TO INITIALIZATION TABLE
40      000000G
41
42      ; SETUP FOR OUTER LOOP
43      000020 005001      CLR  R1                  ; SET COUNTERS
44      000022 005002      CLR  R2
45
46      ; BEGIN OUTER (MONITOR) LOOP -
47      ; RESET STACK POINTER INITILIZOR AND SET UP START ADDRESS TABLE
48
49      000024 016262 INT010: MOV  ISTART(R2), START(R2); STORE START ADDRESS
50      000032 016705      MOV  CTOP, R5            ; CALCULATE LOW ADDRESS OF MONITOR
51      000030

```

INIT ; MACRO V004A PAGE 1+

```

51 00036 166205      SUB REGION(R2),R5 ;
      0000000
52 00042 020567      CMP R5,SPSTRT ;CHECK AGAINST CURRENT LOW ADDRESS
      000322
53 00046 002002      BGE INT020 ;BRANCH IF NOT LESS THAN
54 00050 010567      MOV R5,SPSTRT ;ELSE STORE NEW VALUE
      000314

55 ;
56 ; ALLOCATE SYSTEM TABLE SPACE
57 ;
58 00054 116104 INT020: MOV B INITBL(R1),R4 ;CHECK TO SEE IF THERE ARE OVERLAYS
      0000000
59 00060 002476      BLT INT090 ;BRANCH IF SO
60 00062 110461      MOV B R4,NTABLE(R1) ;SET NUMBER OF OVERLAYS IN NTABLE
      0000000
61 00066 016762      MOV AMODTB,MODTBL(R2);SET BASE ADDRESS OF OVERLAY TABLE
      000300
      0000000
62 00074 005204      INC R4 ;UPDATE BASE ADDRESS
63 00076 010405      MOV R4,R5 ; BA = BA + 6 * NO. OF OVERLAYS
64 00100 006305      ASL R5 ;
65 00102 060405      ADD R4,R5 ;
66 00104 006305      ASL R5 ;
67 00106 060567      ADD R5,AMODTB ;
      000260
68 00112 026727      CMP AMODTB,#MODEND ;CHECK TO SEE IF TABLE LENGTH EXCEEDED
      000254
      0000000
69 00120 003414      BLE INT030 ;BRANCH IF NOT
70 00122 012746      MOV #TABERR,-(SP) ;ELSE PRINT ERROR MESSAGE
      000406
71 00126 012746      MOV #PRLINK,-(SP) ;
      000376
72 00132 104002      EMT 2 ;
73 00134 012746      MOV #PRLINK,-(SP) ;WAIT
      000376
74 00140 104001      EMT 1 ;
75 00142 012746      MOV #PRLINK,-(SP) ;RELEASE TTY
      000376
76 00146 104007      EMT 7 ;
77 00150 104060      EMT 60 ;AND EXIT
78 00152 110167 INT030: MOV B R1,SYSMOD ;PUT MONITOR IN CONTROL
      0000000
79 00156 005003      CLR R3 ;CLEAR OVERLAY NUMBER
      0000000
80 ;
81 ; SET UP TO INITILIZE OVERLAYS
82 ;
83 00160 112005 INT050: MOV B (R0)+,R5 ;GET NUMBER OF BYTES TO FOLLOW
84 00162 112067      MOV B (R0)+,STATUS ;GET STATUS BYTE
      000031
85 00166 001431      BEQ INT080 ;BRANCH IF OVERLAY NOT TO BE INITIALIZED
86 00170 062705      ADD #STATUS+1,R5 ;ELSE SET UP COUNTER FOR MOVE LOOP
      000220
87 00174 012045 INT060: MOV (R0)+,-(R5) ;STORE WORD
88 00176 020527      CMP R5,#STATUS+1 ;DONE?
      000220
89 00202 003374      BGT INT060 ;IF NOT, BRANCH

```

```

INIT          ; MACRO V004A PAGE 1+

90 00204 110367          MOVB R3,OVLNO          ;STORE OVERLAY NUMBER
           000006

91
92          ;
93          ; INITILIZE SYSTEM TABLE
94 00210 004567          JSR R5,UPDATE          ;JUMP TO UPDATE
           000000G
95 00214 000406          BR INT070
96 00216 000          OVLNO: .BYTE 0          ; OVERLAY NUMBER
97 00217 000          STATUS: .BYTE 0          ; STATUS BYTE
98 00220 000000          .WORD 0,0,0,0          ; PARAMETERS
           00222 000000
           00224 000000
           00226 000000
           00230 000000
99 00232 132767 INT070: BITB #200,STATUS          ;CHECK FOR SUCCESSFUL COMPLETION
           000200
           177757
100 0240 001404          BEQ INT080          ;BRANCH IF SO
101 0242 012705          MOV #OVLNO,R5          ;ELSE PRINT ERROR MESSAGE
           000216
102 0246 004767          JSR PC,FILERR          ;
           000000G
103
104          ;
105          ; END OF OVERLAY LOOP
106 0252 005304 INT080: DEC R4          ;DECREMENT COUNTER
107 0254 001341          BNE INT050          ;BRANCH IF NOT DONE
108
109          ;
110          ; END OF MONITOR LOOP
111 0256 005722 INT090: TST (R2)+          ;INCREMENT COUNTERS
112 0260 005201          INC R1
113 0262 120167          CMPB R1,INTABL          ;CHECK FOR COMPLETION
           003000G
114 0266 003002          BGT INT100          ;BRANCH IF SO
115 0270 000167          JMP INT010          ;ELSE RETURN FOR ANOTHER
           177530
116
117          ;
118          ; SET UP FOR SUBSYSTEM LOOP
119 0274 116767 INT100: MOVB NSSYS,NSYS          ;INITILIZE COUNTERS
           000000G
           000000G
120 0302 012700          MOV #ISYSNM,R0
121 0306 012701          MOV #SYSNAM,R1
122 0312 012702          MOV #ISYSTB,R2
123 0316 012703          MOV #SYSTAB,R3
124 0322 012704          MOV #SYSEND,R4
           000000G
           000300G
125
126          ;
127          ; SET UP SUBSYSTEM TABLES

```

INIT ; MACRO V004A PAGE 1+

```

128 0326 112021 INT110: MOV B (<R0>+,<R1>+ ;STORE MONITOR NUMBER
129 0330 012223      MOV (<R2>+,<R3>+ ;STORE SUBSYSTEM NAME
130 0332 020004      CMP R0,R4 ;CHECK TO SEE IF DONE
131 0334 003774      BLE INT110 ;BRANCH IF NOT
132
133 ;
134 ; FINISH INITIALIZATION
135 0336 012746      MOV #PRLINK,-(SP) ;RELEASE TTY
136 0342 104007      EMT 7 ;
137 0344 016767      MOV CTOP,CORTOP ;SET CORTOP
138 0352 105067      CLRB SYSMOD ;PUT SYSTEM IN CONTROL
139 0356 016706      MOV SPSTRT,SP ;RESET STACK POINTER
140 0362 000167      JMP SVC ;JUMP TO SYSTEM
141
142 ;
143 ; DATA ALLOCATIONS
144 0366 075200 CTOP: .WORD 75200 ;TOP OF CORE
145 0370 075200 SPSTRT: .WORD 75200 ;STACK POINTER INITIALIZOR
146 0372 000000 AMODTB: .WORD 0 ;BASE ADDRESS
147
148 ;
149 ; I/O BLOCK AND BUFFER
150 0374 000000 PRLINK: .WORD 0 ;TTY CONTROL BLOCK
151 0376 000000      .WORD 0
152 0400 000000      .WORD 0
153 0402 001      .BYTE 1
154 0403 000      .BYTE 0
155 0404 042420      .RAD50 /KB /
156 0406 000020 TABERR: .WORD 16. ;ERROR MESSAGE
157 0410 000      .BYTE 0
158 0411 000      .BYTE 0
159 0412 000020      .WORD 16.
160 0414 124      .ASCII /TABLE OVERFLOW/
161 0415 101
162 0416 102
163 0417 114
164 0420 105
165 0421 040
166 0422 117
167 0423 126
168 0424 105
169 0425 122
170 0426 106
171 0427 114
172 0430 117
173 0431 127
174 0432 015      .BYTE CR,LF
175 0433 012
176 000000      .END INIT

```

INIT ; MACRO V004A PAGE 1+
SYMBOL TABLE

AMODTB 000372R	CORTOP= ***** G	CR = 000015
CTOP 000366R	FILERR= ***** G	INIT 000000R
INITBL= ***** G	INTABL= ***** G	INT010 000024R
INT020 000054R	INT030 000152R	INT050 000160R
INT060 000174R	INT070 000232R	INT080 000252R
INT090 000256R	INT100 000274R	INT110 000326R
ISTART= ***** G	ISYSNM= ***** G	ISYSTB= ***** G
LF = 000012	MODEND= ***** G	MODTBL= ***** G
NSSYS = ***** G	NSYS = ***** G	NTABLE= ***** G
OVLNO 000216R	PC =%000007	PRLINK 000376RG
REGION= ***** G	R0 =%000000	R1 =%000001
R2 =%000002	R3 =%000003	R4 =%000004
R5 =%000005	SP =%000006	SPSTRT 000370R
START = ***** G	STATUS 000217R	SVC = ***** G
SYSEND= ***** G	SYSMOD= ***** G	SYSNAM= ***** G
SYSTAB= ***** G	TABERR 000406R	UPDATE= ***** G

. ABS. 000000 000
000434 001

ERRORS DETECTED: 0
FREE CORE: 4454. WORDS
INIT.BIN,INIT.LST/CRF<DT0:INIT.SRC

CROSS REFERENCE TABLE S-1

AMODTB	1- 38@	1- 61	1- 67@	1- 68	1-146#				
CORTOP	1- 18#	1-137@							
CR	1- 31#	1-161							
CTOP	1- 50	1-137	1-144#						
FILERR	1- 18#	1-102							
INIT	1- 36#	1-162							
INITBL	1- 16#	1- 39	1- 58						
INTABL	1- 16#	1-113							
INT010	1- 49#	1-115							
INT020	1- 53	1- 58#							
INT030	1- 69	1- 78#							
INT050	1- 83#	1-107							
INT060	1- 87#	1- 89							
INT070	1- 95	1- 99#							
INT080	1- 85	1-100	1-106#						
INT090	1- 59	1-111#							
INT100	1-114	1-119#							
INT110	1-128#	1-131							
ISTART	1- 16#	1- 49							
ISYSNM	1- 16#	1-120							
ISYSTB	1- 16#	1-122							
LF	1- 32#	1-161							
MODEND	1- 18#	1- 68							
MODTBL	1- 18#	1- 38	1- 61@						
NSSYS	1- 16#	1-119							
NSYS	1- 17#	1-119@							
NTABLE	1- 17#	1- 60@							
OVLNO	1- 90@	1- 96#	1-101						
PC	1- 27#	1-102@							
PRLINK	1- 18#	1- 36	1- 71	1- 73	1- 75	1-135	1-151#		
REGION	1- 16#	1- 51							
R0	1- 20#	1- 39@	1- 83	1- 84	1- 87	1-120@	1-128	1-130	
R1	1- 21#	1- 43@	1- 58	1- 60@	1- 78	1-112@	1-113	1-121@	
	1-128@								
R2	1- 22#	1- 44@	1- 49@	1- 51	1- 61@	1-111	1-122@	1-129	
R3	1- 23#	1- 79@	1- 90	1-123@	1-129@				
R4	1- 24#	1- 58@	1- 60	1- 62@	1- 63	1- 65	1-106@	1-124@	
	1-130								
R5	1- 25#	1- 50@	1- 51@	1- 52	1- 54	1- 63@	1- 64@	1- 65@	
	1- 66@	1- 67	1- 83@	1- 86@	1- 87@	1- 88	1- 94@	1-101@	
SP	1- 26#	1- 36@	1- 70@	1- 71@	1- 73@	1- 75@	1-135@	1-139@	
SPSTRT	1- 52	1- 54@	1-139	1-145#					
START	1- 17#	1- 49@							
STATUS	1- 84@	1- 86	1- 88	1- 97#	1- 99				
SVC	1- 18#	1-140							
SVSEND	1- 16#	1-124							
SYSMOD	1- 18#	1- 78@	1-138@						
SYSNAM	1- 17#	1-121							
SVSTAB	1- 17#	1-123							
TABERR	1- 70	1-156#							
UPDATE	1- 18#	1- 94							

APPENDIX B

AN EXAMPLE OF THE TABLE MODULE

```

1
2      : MODULE NAME      INITIALIZATION TABLES
3
4      : AUTHOR/DATE      DAVID MUELLER      JANUARY, 1974
5
6      : MACHINE/LANGUAGE  PDP-11/20, 16K      PAL-11R  V005A
7
8      : PURPOSE          TABLES FOR USE BY THE SYSTEM PRODUCTION INITIALIZOR
9
10     :
11     : SPECIFICATION STATEMENTS
12     000000 000000 6      :
13     : .TITLE TABLES      ;
14     : .GLOBL INITBL, ISTART, NSSYS, ISYSNM, ISYSTB, SYSEND, REGION
15     000002      : .CSECT      ;
16
17     : SUBSYSTEM TABLES AND VARIABLES
18     : SUBSYSTEM NAMES (IN RADIX 50)
19
20     000002 027221 ISYSNM: .RAD50 /GRA/      ; SUBSYSTEM 0
21     000004 035054      .RAD50 /ILL/      ; SUBSYSTEM 1
22     000006 000000      .WORD 0      ; SUBSYSTEM 2
23     000010 000000      .WORD 0      ; SUBSYSTEM 3
24     000012 000000      .WORD 0      ; SUBSYSTEM 4
25
26     :
27     : NUMBER OF SUBSYSTEMS
28     000014 002 NSSYS: .BYTE 2      ;
29
30     : MONITOR TABLE
31
32     000015 002 ISYSTB: .BYTE 2      ; SUBSYSTEM 0
33     000016 004      .BYTE 4      ; SUBSYSTEM 1
34     000017 377      .BYTE -1      ; SUBSYSTEM 2
35     000020 377      .BYTE -1      ; SUBSYSTEM 3
36     000021 377 SYSEND: .BYTE -1      ; SUBSYSTEM 4
37
38     : MO % 0      50
39     : NUMBER OF OVERLAYS / MONITOR TABLE - 1 BYTE / MONITOR CONTAINING
40     : (NUMBER OF OVERLAYS - 1)
41
42     000022 005 INTABL: .BYTE 5      ; SYSTEM = NUMBER OF MONITORS + 1
43     000023 377      .BYTE -1      ; DUMMY MONITOR
44     000024 377      .BYTE -1      ; GRAFIX COMPILER
45     000025 001      .BYTE 1      ; BASIC PLOTTING PACKAGE MONITOR
46     000026 000      .BYTE 0      ; ILLISYM
47     000027 003      .BYTE 3      ; DIFFSUB
48
49     :
50     : STARTING ADDRESSES TABLE
51
52     000030 072632 ISTART: .EVEN      ;
53     000032 073412      .WORD 072632      ; LINE PRINTER
54     000034 071504      .WORD 073412      ; ERROR
55     000036 073516      .WORD 071504      ; GRAFIX COMPILER
56     000040 062504      .WORD 073516      ; BASIC PLOTTING PACKAGE
57     000042 074100      .WORD 062504      ; ILLISYM
58     : DIFFSUB

```

```

58
59
60
61
62
63 00044 007072 REGION: .WORD 007072 ;LINE PRINTER
64 00046 001565 .WORD 001565 ;ERROR
65 00050 003472 .WORD 003472 ;GRAFIX COMPILER
66 00052 005217 .WORD 005217 ;BASIC PLOTTING PACKAGE
67 00054 012472 .WORD 012472 ;B
68 00056 012472 .WORD 012472 ;ILLISYM
69 00060 015704 .WORD 015704 ;DIFFSUB
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102 0062 012 INITBL: .BYTE 12 ;MONITORS
103 0063 130 .BYTE 130 ;LINE PRINTER
104 0064 003435 .WORD 003435 ;NUMBER OF WORDS TO FOLLOW
105 0066 000502 .RAD50 /OVR/ ;STATUS BYTE
106 0070 000000 .RAD50 / / ;LENGTH
107 0072 046600 .RAD50 /LP / ;EXTENSION
108 0074 012 .BYTE 12 ;FILE NAME
109 0075 130 .BYTE 130 ;ERROR
110 0076 000672 .WORD 000672
111 0100 073412 .WORD 073412
112 0102 060502 .RAD50 /OVR/
113 0104 060220 .RAD50 /OR /
114 0106 021042 .RAD50 /ERR/

```

REGION SIZE TABLE - 1 ENTRY / MONITOR GIVING THE LARGEST AREA OF CORE (IN BYTES) EVER OCCUPIED BY THE MONITOR AND ITS OVERLAYS

OVERLAY TABLE - ALL ENTRIES ARE THE SAME

- ONE ENTRY FOR EACH POSSIBLE MONITOR AND OVERLAY NUMBER
- ALL MONITORS ARE LISTED FIRST, FOLLOWED BY OVERLAYS IN THE ORDER OF THEIR MONITORS
- ENTRY FORMAT
 - I) THE FOLLOWING PARAMETERS ARE REQUIRED, IN THE ORDER INDICATED, 1 BYTE / PARAMETER
 - 1) NUMBER OF BYTES TO FOLLOW - NUMBER OF BYTES OF OPTIONAL PARAMETERS
 - 2) STATUS BYTE - SET UP AS INDICATED IN THE DOCUMENTATION TO ROUTINE UPDATE
 THESE TWO ARE BOTH EQUAL TO ZERO FOR MONITORS AND OVERLAYS WHICH ARE NOT TO BE INITIALIZED
 - II) THE FOLLOWING PARAMETERS ARE OPTIONAL, AND ARE PRESENT AS REQUIRED BY THE STATUS BYTE. ALL ARE FULLWORD PARAMETERS AND MUST BE IN THE ORDER GIVEN IF THEY ARE PRESENT
 - 1) LENGTH OF MONITOR OR OVERLAY, IN WORDS
 - 2) MEMORY ADDRESS OF START OF MONITOR OR OVERLAY
 - 3) EXTENSION OF FILE NAME WHERE MONITOR OR OVERLAY IS LOCATED, IN RADIX 50
 - 4) FINAL THREE CHARACTERS OF FILE NAME OF FILE WHERE MONITOR OR OVERLAY IS LOCATED IN RADIX 50
 - 5) INITIAL THREE CHARACTERS OF FILE NAME OF FILE WHERE MONITOR OR OVERLAY IS LOCATED IN RADIX 50

115	0110	012	.BYTE 12
116	0111	130	.BYTE 130
117	0112	001635	.WORD 001635
118	0114	071504	.WORD 071504
119	0116	051646	.RAD50 /MON/
120	0120	062564	.RAD50 /PIL/
121	0122	012445	.RAD50 /COM/
122	0124	012	.BYTE 12
123	0125	130	.BYTE 130
124	0126	001160	.WORD 001160
125	0130	072636	.WORD 072636
126	0132	051646	.RAD50 /MON/
127	0134	000000	.RAD50 / /
128	0136	007420	.RAD50 /BPP/
129	0140	012	.BYTE 12
130	0141	130	.BYTE 130
131	0142	005235	.WORD 005235
132	0144	062504	.WORD 062504
133	0146	051646	.RAD50 /MON/
134	0150	075265	.RAD50 /SYM/
135	0152	035054	.RAD50 /ILL/
136	0154	012	.BYTE 12
137	0155	130	.BYTE 130
138	0156	004000	.WORD 004000
139	0160	065176	.WORD 065176
140	0162	051646	.RAD50 /MON/
141	0164	075012	.RAD50 /SUB/
142	0166	015156	.RAD50 /DIF/
143			
144			
145	0170	012	.BYTE 12
146	0171	130	.BYTE 130
147	0172	067757	.WORD 067757
148	0174	060474	.RAD50 /OVL/
149	0176	004505	.RAD50 /ASM/
150	0200	007420	.RAD50 /BPP/
151	0202	012	.BYTE 12
152	0203	130	.BYTE 130
153	0204	004500	.WORD 004500
154	0206	062757	.WORD 062757
155	0210	060474	.RAD50 /OVL/
156	0212	021403	.RAD50 /EXC/
157	0214	007420	.RAD50 /BPP/
158			
159	0216	012	.BYTE 12
160	0217	130	.BYTE 130
161	0220	001235	.WORD 001235
162	0222	062504	.WORD 062504
163	0224	060474	.RAD50 /OVL/
164	0226	075265	.RAD50 /SYM/
165	0230	035054	.RAD50 /ILL/
166			
167	0232	000	.BYTE 0
168	0233	000	.BYTE 0
169	0234	012	.BYTE 12
170	0235	130	.BYTE 130
171	0236	000640	.WORD 000640

GRAFIX COMPILER

BASIC PLOTTING PACKAGE MONITOR

ILLISYM

DIFFSUB

OVERLAYS
BASIC PLOTTING PACKAGE
OVERLAY 0

OVERLAY 1

ILLISYM
OVERLAY 0

DIFFSUB
OVERLAY 0

OVERLAY 1

TABLES ; MACRO V004R PAGE 1+

```

172 0240 057272 .WORD 057272
173 0242 060517 .RAD50 /OVI/
174 0244 075012 .RAD50 /SUB/
175 0246 015156 .RAD50 /DIF/
176 0250 012 .BYTE 12
177 0251 130 .BYTE 130
178 0252 000250 .WORD 000250
179 0254 065176 .WORD 065176
180 0256 060520 .RAD50 /OV2/
181 0260 075012 .RAD50 /SUB/
182 0262 015156 .RAD50 /DIF/
183 0264 012 .BYTE 12
184 0265 130 .BYTE 130
185 0266 001235 .WORD 001235
186 0270 062504 .WORD 062504
187 0272 060474 .RAD50 /OVL/
188 0274 075265 .RAD50 /SYM/
189 0276 035054 .RAD50 /ILL/
190
191 .END OF TABLES
192
193 000001 .END

```

OVERLAY 2

OVERLAY 3

TABLES ; MACRO V004R PAGE 1+
SYMBOL TABLE

```

INITBL 000062RG INTABL 000022R ISTART 000030RG
ISYSNM 000002RG ISYSTB 000015RG NSSYS 000014RG
REGION 000244RG SYSEND 000021RG
. ABS. 000000 000
000300 001

```

ERRORS DETECTED: 0
FREE CORE: 5050. WORDS
TABLE.BIN, TABLE.LST/CRF<TABLE.SRC

CROSS REFERENCE TABLE S-1

```

INITBL 1- 14# 1-102#
INTABL 1- 42#
ISTART 1- 14# 1- 52#
ISYSNM 1- 14# 1- 20#
ISYSTB 1- 14# 1- 32#
NSSYS 1- 14# 1- 28#
REGION 1- 14# 1- 63#
SYSEND 1- 14# 1- 36#

```

Bibliography

Brown, R. Leonard, Jr., Numerical Systems on A Mnicomputer, Report No. UIUCDCS-R-73-555 AEC-COO-1469-0215, Department of Computer Science, University of Illinois, Urbana, February, 1973.

Digital Equipment Corporation, PDP 11-20 Processor Handbook, Maynard, Massachusetts, 1971.

Mueller, David, A Batch System with Rapid Overlay Capabilities, Report No. UIUCDCS-R-73-614 AEC-COO-2383-0005, Department of Computer Science, University of Illinois, Urbana, January, 1974.

Mueller, David, The ICGS System: Users Manual, Report No. UIUCDCS-R-73-608 AEC-COO-2383-0002, Department of Computer Science, University of Illinois, Urbana, January, 1974.

BIBLIOGRAPHIC DATA HEET		1. Report No. UIUCDCS-R-74-608	2.	3. Recipient's Accession No.	
Title and Subtitle THE ICGS SYSTEM: USERS MANUAL				5. Report Date February 1974	
				6.	
7. Author(s) David Henry Meuller				8. Performing Organization Rept. No.	
9. Performing Organization Name and Address Department of Computer Science University of Illinois Urbana, Illinois 61801				10. Project/Task/Work Unit No.	
				11. Contract/Grant No. US AEC AT(11-1)2383	
12. Sponsoring Organization Name and Address US AEC Chicago Operations Office 9800 South Cass Avenue Argonne, Illinois				13. Type of Report & Period Covered Thesis Research	
				14.	
15. Supplementary Notes					
16. Abstracts The Illinois Computing Graphics System (ICGS) is a batch system for a minicomputer. It executes a limited number of system programs, and provides facilities for rapid overlaying of these programs. This manual is intended to aid the user in preparing system modules and executing the system.					
17. Key Words and Document Analysis. 17a. Descriptors Supervisory Systems Executive routines overlays minicomputers					
17b. Identifiers/Open-Ended Terms					
c. COSATI Field/Group					
18. Availability Statement unlimited			19. Security Class (This Report) UNCLASSIFIED		21. No. of Pages 16
			20. Security Class (This Page) UNCLASSIFIED		22. Price