N69-30039
NASA CR-25823
NAS 12-140
CR-86154

E-2410

SUMMARY REPORT
ON-LINE LOGICAL SIMULATION (OLLS)

by

H. R. Howie, G. Schwartz,
and
H. A. Thaler

April 1969

CASE FILE
COPY

INSTRUMENTATION
LABORATORY

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

Cambridge 39, Mass.

E-2410

SUMMARY REPORT

ON-LINE LOGICAL SIMULATION (OLLS)

by

H.R. Howie, G. Schwartz,
and
H.A. Thaler

April 1969

INSTRUMENTATION LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY
CAMBRIDGE, MASSACHUSETTS

Approved: Eldon C Hall Date: 13 May 1969
E.G. HALL, DIRECTOR, DIGITAL DEVELOPMENT
INSTRUMENTATION LABORATORY

Approved: Ralph R. Ragan Date: 14 May 1969
R.R. RAGAN, DEPUTY DIRECTOR
INSTRUMENTATION LABORATORY

ACKNOWLEDGMENT

This report was prepared under DSR Project 55-27600, sponsored by the Electronics Research Center of the National Aeronautics and Space Administration through Contract NAS 12-140 with the Instrumentation Laboratory of Massachusetts Institute of Technology in Cambridge, Massachusetts.

The publication of this report does not constitute approval by the National Aeronautics and Space Administration of the findings or the conclusions contained therein. It is published only for the exchange and stimulation of ideas.

E-2410

Summary Report
ON-LINE LOGICAL SIMULATION (OLLS)

ABSTRACT

This report is intended to summarize the progress in implementing the system described in an earlier document MIT/IL Report E-2265, ON-LINE LOGICAL SIMULATION (OLLS), written in May 1968. The topics discussed in that document will be reported here in the same order. It is assumed that the reader is familiar with the contents of E-2265 and no attempt will be made here to redescribe the technical makeup of the various subsystems. Results of computer test runs and the to-date status of the programs are presented.

by H. R. Howie
G. Schwartz
H. A. Thaler
April 1969

TABLE OF CONTENTS

Section		Page
1.	INTRODUCTION	1
2.	THE MAC 360 WORKING SYSTEM	5
3.	THE FILLIP CARD INPUT SYSTEM	9
	3.1 General	9
	3.2 Existing Programs	9
	3.3 Contemplated Programs	28
4.	THE CRT INTERACTIVE SYSTEM	29
5.	CONCLUSION	67

1. INTRODUCTION

The possibility of using computers to aid designers has been recognized and exploited, in various ways, for the last several years. Designers can have mechanized help in small circuit design (ECAP, NET), and in some forms of mechanical design (SKETCHPAD). As SKETCHPAD showed, the implications of a cathode ray tube system whereby the designer and the computer interact, as opposed to the more prevalent processing systems, are many and exciting.

The possibility of using major data processing aids for logical designs became an important concern to those who had been engaged, for quite some time, in the development of medium-sized computer systems, especially if those systems could be made interactive. But interactive or not, accumulated experience in logical design indicated the near necessity of mechanized files, drafting aids, and simulations.

The initial objective of MIT/IL was not so much to demonstrate the power of a new approach (Computed Aided Design) as to develop and implement a practical system. We are still short of that goal in that we do not have an operational interactive system; we do have a batch system (MAC 360) and major portions of the more ambitious FILLIP List Processing System and the CRT Interactive System. Consequently, the present report is in part a demonstration of achievement and in part a blueprint of present and future developments.

MAC 360 is a card system, with very limited file capability, wherein the logical device models are an integral part of the program. It was written without recourse to a list-processing language, and has been in use for about six months.

In early 1967 a decision was made by the Digital Computation group (which runs the data processing system of Instrumentation Laboratory) to implement a major list-processing language called FILLIP, and it was decided then that OLLS/360 should be based on FILLIP. As of this writing, FILLIP is still under development for its overall system aspects, and, consequently, some of the FILLIP Card System remains untested. The major features of FILLIP, and its power, are described in The Users' Guide to FILLIP by Charles A. Muntz and J. Halcombe Laning, Jr.

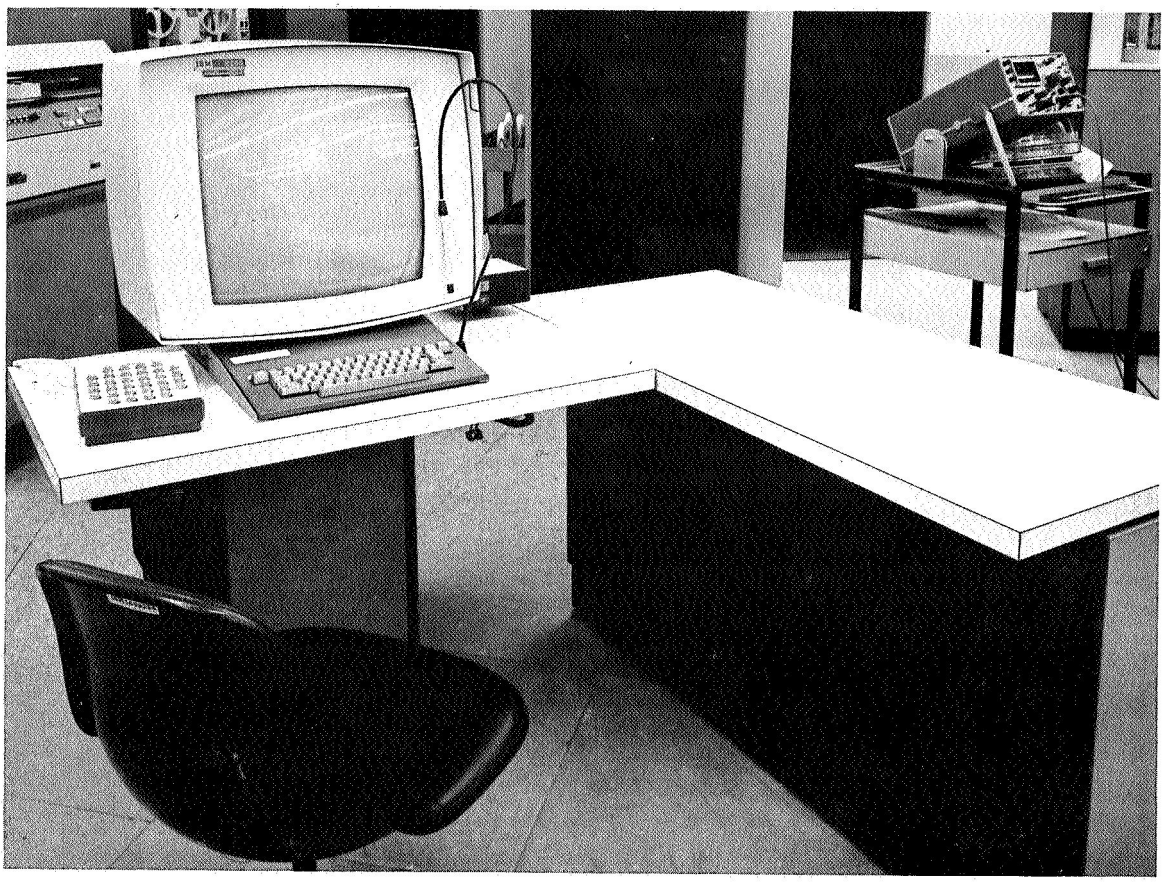


Fig. 1-1 OLLS/360 Installation

When it became apparent that the FILLIP language would not be operational before mid 1968, and FILLIP would not be able to handle difficult I/O such as the CRT interface or a plotter, we decided to implement a machine language version of the logic file specifically with an interactive CRT - operator interface in mind. While this version may lack some of the elegance present in the FILLIP version made possible by such FILLIP operations as "TREE", "SICS", and "MDUPL" which allow large file structures to be searched and manipulated by only a few lines of source coding, similar operations are not difficult to implement in machine language. The tradeoffs are much more source coding for speed, efficiency, and the ability to access any I/O device attached to the CPU.

2. THE MAC 360 WORKING SYSTEM

This system is very similar to the so-called H1800 system described in Chapter 2 of E-2265. The original programs (written in a language called MAC) were translated and expanded from the versions which ran on the Honeywell 1800 to be a useful design tool on the IBM 360. With the exception of a few input-output routines which are written in machine language, the IBM 360 version is also written in the language of MAC.

Many changes and improvements were made in the system.

a) The list of available device types was expanded to 34 devices.

b) The running time for a typical drawing was reduced to about 1.5 minutes of CPU time.

c) The REVISE mode was removed from DRAWSCHEMATIC and replaced by a separate program which can rename devices and signals if required.

d) The aesthetic quality of the output plots was greatly improved by the addition of a routine which puts a frame on the plot which conforms to standard drafting conventions. (See Fig. 2-1)

The MAC 360 system is in full production use at MIT/IL, and, although it was never intended to be used in production, we feel we have learned much from the experience.

a) Since most logic designers are unaccustomed to punching cards, a card input system can be a very frustrating experience for the designer at first. Too much time and energy is spent learning and conforming to card formats and trivial details, and too little time is spent in design.

b) A system with no immediate feedback to the designer is very slow and costly. Most drawings require 3 or 4 reruns before they become a finished part of the logic design. This implies a process time of about one week and a cost of about 6 CPU minutes per drawing.

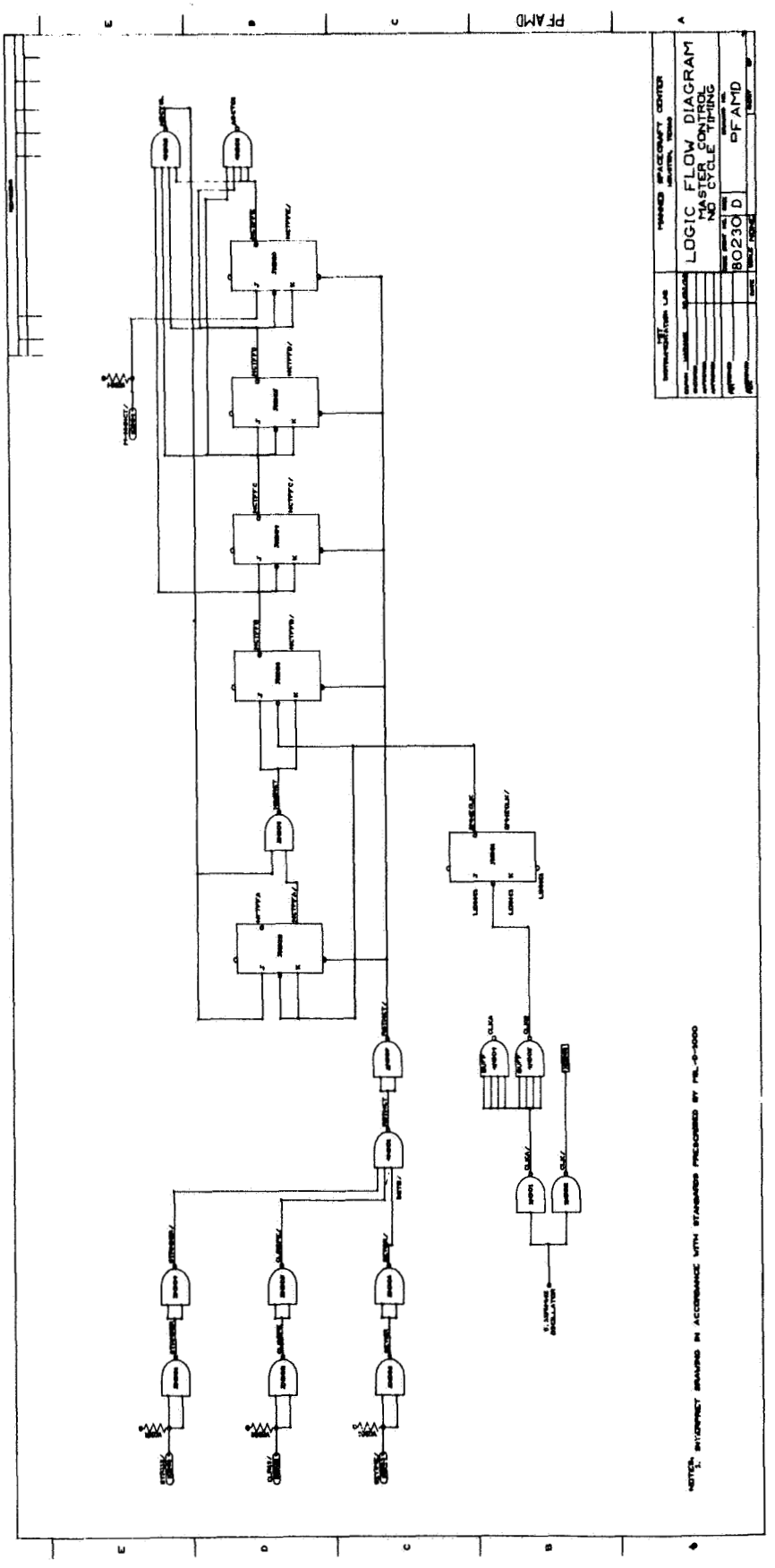


Fig. 2-1 Drawing Format

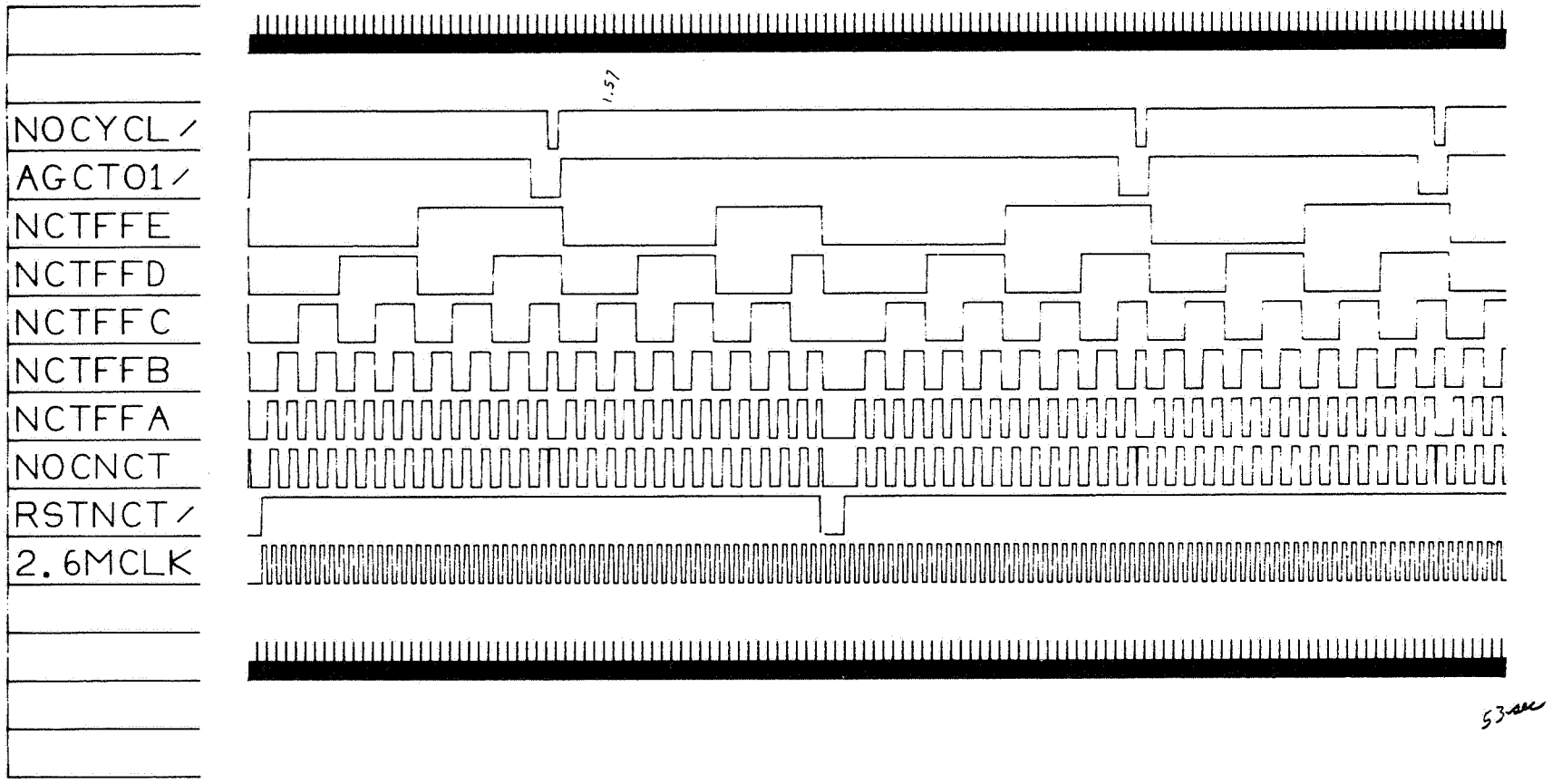


Fig. 2-2 PFAMD Simulation

For most logic systems which will have many such drawings, the effect on schedules and budgets is severe. Most of the difficulty is due to the fact that the entire drawing must be resubmitted in order to make even minor corrections. The former REVISE mode was no better since even though fewer input cards were submitted for a revision, the program still had to load up the entire old file to process a revision. An interactive system of even the most rudimentary type would be easier to learn, faster to operate (both CPU and real time), and place the designer in a position where he is not completely at the mercy of a batch program.

c) The inflexibility of this program keeps the designer in a very tight box. He is constantly faced by problems such as not enough device types, not enough space on the drawing, small logic file, etc.

What we learned most from this system is that designers dislike IBM cards. They also have little use for printed computer output except for finished signal lists (and even then only if someone else will be reading the list). Supervisors are uneasy about designers who must spend too much time doing non-design type clerical work, and they are especially uneasy about high costs and schedule bottlenecks.

The system does work however, and its main selling point is the ability to produce and maintain accurate signal lists and wire wrap control cards, and as such it will remain useful (if cumbersome and costly) until an interactive system is operational.

Figure 2-1 shows the new drawing format. We include it here for its own sake and to aid in the interpretation of the computer output for the creation and simulation of PFAMD shown in Section 3. (Figs. 3-1 through 3-16).

The simulation of PFAMD shown in Fig. 2-2 was obtained by injecting inputs to the circuit at "2.6MCLK" and "RSTNCT/". Note the appearance of a "sneak pulse" on signal "NOCNCT" coincident with the falling edge of signal "NOCYCL/". This represents an essential circuit hazard and is also discovered by the FILLIP simulation. (See Fig. 3-14 at time 297 and Fig. 3-15)

3. THE FILLIP CARD INPUT SYSTEM

3.1 General

The status of the OLLS card system is, as it has been in the past, closely related to that of the FILLIP language. A recent milestone in the development of this language was the implementation of a "stored file" capability, which allows for the creation, deletion, calling, and storage (on disc) of FILLIP files. Now it is possible to save data files between runs and to store programs, which correspond to OLLS asterisk cards, in separate files.

Of course, this state of FILLIP's development was not unexpected; we have always endeavored to write our programs in a way which would require no conversion when the "stored file" version of FILLIP became available. Only the CARDREAD program, which does the actual file manipulation, needed rewriting; this has been done, but the new program has not been debugged.

The new CARDREAD program treats a particular data file either as a read-only (henceforth called a "read") file or as a read/write ("write" or "working") file. There is nothing inherent to a FILLIP file which makes it a read or a write file; this distinction is made by OLLS to enable its users to use the same file for several asterisk instructions without repeating its name on each card, and to simplify the syntax of instructions which require several variables.

Figures 3-1 through 3-16 show the printer output from a recent FILLIP OLLS run in which a data file was created, three devices were defined, and a circuit, essentially the same as PFAMD shown in Fig. 2-1, was constructed and simulated. The simulation output from the printer corresponds to the plotted output from the MAC 360 simulation in Fig. 2-2.

3.2 Existing Programs

The following asterisk instructions are either working or are being debugged:

* OPEN (3.2.1)
* DEFINE (3.2.2)

END OF STEP * CREATE FILE PFAMO

Figure 3-1

```
        OUTPUTS A
        INPUTS B C D E
        DEFAULT 1 1 1 1
        EQUATION A = ¬ ( B * C * D * E )
        FORMAT A B C D E


END OF STEP     * DEFINE DEVICE NAND
```


Figure 3-2

```
OUTPUTS A B
INPUTS J K T S R
DEFAULT 1 1 1 1 1
EQUATION A = ¬ S + T * ¬ T * ( J * ¬ A + ¬ K * A ) + ¬ ( T * ¬ T ) * ¬ B
DELAY          1   2     1     1     1     1   1         2     1       1
EQUATION B = ¬ R + T * ¬ T * ( K * ¬ B + ¬ J * B ) + ¬ ( T * ¬ T ) * ¬ A
DELAY          1   2     1     1     1     1   1         2     1       1
FORMAT A B J K T S R
```

END OF STEP * DEFINE DEVICE JK

Figure 3-3

```
        OUTPUTS B
        INPUTS A
        EQUATION B = A * ¬ B
        DELAY 0 5
        FORMAT B A

    END OF STEP    * DEFINE DEVICE OSC
```


Figure 3-4

```
      NAND 1 1 1 1 RSTNCT/ RSTNCT
      NAND 2 1 2 2 NOCNCT NOCYCL/ NCTFFA/
      NAND 3 1 3 3 NOCYCL/ NCTFFB NCTFFC NCTFFD NCTFFE
      NAND 4 1 4 4 AGCTO1/ NCTFFC NCTFFD NCTFFE
      NAND 10 1 20 20 N10  N12
      NAND 11 1 21 21 N11 N10
      NAND 12 1 22 22 N12 N11
      JK 1 1 6 6 NCTFFA NCTFFA/ NOCYCL/ $  CLOCK $  RSTNCT/
      JK 2 1 7 7 NCTFFB NCTFFB/ NOCNCT NOCNCT CLOCK $  RSTNCT/
      JK 3 1 8 8 NCTFFC NCTFFC/ $ $ NCTFFB $  RSTNCT/
      JK 4 1 9 9 NCTFFD NCTFFD/ $ $ NCTFFC $  RSTNCT/
      JK 5 1 10 10 NCTFFE NCTFFE/ $ $ NCTFFD $  RSTNCT/
```



```
END OF STEP    * GET PFAMD
```

Figure 3-5

```
      INITIAL1 RSTNCT

      INITIALO CLOCK

      INITIALO N10

      PROP

SIGNAL  N10      Became Undefined During Propagation, Value Was  0
SIGNAL  N11                                                      1
SIGNAL  N12                                                      0
      PRINT SIGNALS
```

SIGNAL NAME	SOURCE GATE	TERMINAL	VALUE	TIME
RSTNCT/	1	A	0	-524 288
RSTNCT			1	-524 288
NOCNCT	2	A	0	-524 288
NCTFFA/	1	B	1	-524 288
AGCTO1/	4	A	1	-524 288
NCTFFA	1	A	0	-524 288
CLOCK			0	-524 288
NCTFFB	2	A	0	-524 288
NCTFFC	3	A	0	-524 288
NCTFFB/	2	B	1	-524 288
NCTFFD	4	A	0	-524 288
NCTFFC/	3	B	1	-524 288
NCTFFE	5	A	0	-524 288
NCTFFD/	4	B	1	-524 288
NCTFFE/	5	B	1	-524 288

Figure 3-6

NOCYCL/	3	A	1	-524 288
N10	10	A		-524 288
N12	12	A		-524 288
N11	11	A		-524 288
UNUSED1			1	-524 288

```
      MINPULSE 3 3
      EVENT RSTNCT 0 0
      TRACE NOCNCT   CLOCK RSTNCT/ NCTFFA NCTFFB NCTFFC NCTFFD NCTFFE NOCYCL/
      C AGCT01/
      SEQUENCE CLOCK 1 10 0 5
      RUN 350
      PRINT SIGNALS
```

SIGNAL NAME	SOURCE GATE	TERMINAL	VALUE	TIME
RSTNCT/	1	A	1	1
RSTNCT			0	0
NOCNCT	2	A	0	347
NCTFFA/	1	B	1	346
AGCT01/	4	A	1	308
NCTFFA	1	A	0	346
CLOCK	SEQINST	SEQOUT	1	350
NCTFFB	2	A	0	346
NCTFFC	3	A	1	347
NCTFFB/	2	B	1	346

Figure 3-7

NCTFFD	4	A	0	308
NCTFFC/	3	B	0	347
NCTFFE	5	A	0	309
NCTFFD/	4	B	1	308
NCTFFE/	5	B	1	309
NOCYCL/	3	A	1	307
N10	10	A		-524 288
N12	12	A		-524 288
N11	11	A		-524 288
UNUSED1			1	-524 288

```
      PRINT LINTRACE 1
```

17

****

CLOCK		RSTNCT/	NCTFFA	NOCNCT	NCTFFB	NCTFFC	NCTFFD	NCTFFE	AGCTO1/	NOCYCL/
0	1									
1	1	1								
2	1	1								
3	1	1								
4	1	1								
5	0	1								
6	0	1	1							
7	0	1	1	1						
8	0	1	1	1						
9	0	1	1	1						
10	1	1	1	1						
11	1	1	1	1						
12	1	1	1	1						
13	1	1	1	1						
14	1	1	1	1						
15	0	1	1	1						
16	0	1	0	1	1					
17	0	1	0	0	1					
18	0	1	0	0	1					

Figure 3-8

19	0	1	0	0	1
20	1	1	0	0	1
21	1	1	0	0	1
22	1	1	0	0	1
23	1	1	0	0	1
24	1	1	0	0	1
25	0	1	0	0	1
26	0	1	1	0	1
27	0	1	1	1	1
28	0	1	1	1	1
29	0	1	1	1	1
30	1	1	1	1	1
31	1	1	1	1	1
32	1	1	1	1	1
33	1	1	1	1	1
34	1	1	1	1	1
35	0	1	1	1	1
36	0	1	0	1	0
37	0	1	0	0	0	1
38	0	1	0	0	0	1
39	0	1	0	0	0	1
40	1	1	0	0	0	1
41	1	1	0	0	0	1
42	1	1	0	0	0	1
43	1	1	0	0	0	1
44	1	1	0	0	0	1
45	0	1	0	0	0	1
46	0	1	1	0	0	1
47	0	1	1	1	0	1
48	0	1	1	1	0	1
49	0	1	1	1	0	1
50	1	1	1	1	0	1
51	1	1	1	1	0	1
52	1	1	1	1	0	1
53	1	1	1	1	0	1
54	1	1	1	1	0	1
55	0	1	1	1	0	1
56	0	1	0	1	1	1
57	0	1	0	0	1	1
58	0	1	0	0	1	1
59	0	1	0	0	1	1
60	1	1	0	0	1	1
61	1	1	0	0	1	1
62	1	1	0	0	1	1
63	1	1	0	0	1	1
64	1	1	0	0	1	1
65	0	1	0	0	1	1
66	0	1	1	0	1	1
67	0	1	1	1	1	1
68	0	1	1	1	1	1
69	0	1	1	1	1	1

Figure 3-9

70	1	1	1	1	1	1	
71	1	1	1	1	1	1	
72	1	1	1	1	1	1	
73	1	1	1	1	1	1	
74	1	1	1	1	1	1	
75	0	1	1	1	1	1	
76	0	1	0	1	0	1	
77	0	1	0	0	0	0	
78	0	1	0	0	0	0	1
79	0	1	0	0	0	0	1
80	1	1	0	0	0	0	1
81	1	1	0	0	0	0	1
82	1	1	0	0	0	0	1
83	1	1	0	0	0	0	1
84	1	1	0	0	0	0	1
85	0	1	0	0	0	0	1
86	0	1	1	0	0	0	1
87	0	1	1	1	0	0	1
88	0	1	1	1	0	0	1
89	0	1	1	1	0	0	1
90	1	1	1	1	0	0	1
91	1	1	1	1	0	0	1
92	1	1	1	1	0	0	1
93	1	1	1	1	0	0	1
94	1	1	1	1	0	0	1
95	0	1	1	1	0	0	1
96	0	1	0	1	1	0	1
97	0	1	0	0	1	0	1
98	0	1	0	0	1	0	1
99	0	1	0	0	1	0	1
100	1	1	0	0	1	0	1
101	1	1	0	0	1	0	1
102	1	1	0	0	1	0	1
103	1	1	0	0	1	0	1
104	1	1	0	0	1	0	1
105	0	1	0	0	1	0	1
106	0	1	1	0	1	0	1
107	0	1	1	1	1	0	1
108	0	1	1	1	1	0	1
109	0	1	1	1	1	0	1
110	1	1	1	1	1	0	1
111	1	1	1	1	1	0	1
112	1	1	1	1	1	0	1
113	1	1	1	1	1	0	1
114	1	1	1	1	1	0	1
115	0	1	1	1	1	0	1
116	0	1	0	1	0	0	1
117	0	1	0	0	0	1	1
118	0	1	0	0	0	1	1
119	0	1	0	0	0	1	1
120	1	1	0	0	0	1	1

Figure 3-10

121	1	1	0	0	0	1	1
122	1	1	0	0	0	1	1
123	1	1	0	0	0	1	1
124	1	1	0	0	0	1	1
125	0	1	0	0	0	1	1
126	0	1	1	0	0	1	1
127	0	1	1	1	0	1	1
128	0	1	1	1	0	1	1
129	0	1	1	1	0	1	1
130	1	1	1	1	0	1	1
131	1	1	1	1	0	1	1
132	1	1	1	1	0	1	1
133	1	1	1	1	0	1	1
134	1	1	1	1	0	1	1
135	0	1	1	1	0	1	1
136	0	1	0	1	1	1	1
137	0	1	0	0	1	1	1
138	0	1	0	0	1	1	1
139	0	1	0	0	1	1	1
140	1	1	0	0	1	1	1
141	1	1	0	0	1	1	1
142	1	1	0	0	1	1	1
143	1	1	0	0	1	1	1
144	1	1	0	0	1	1	1
145	0	1	0	0	1	1	1
146	0	1	1	0	1	1	1
147	0	1	1	1	1	1	1
148	0	1	1	1	1	1	1
149	0	1	1	1	1	1	1
150	1	1	1	1	1	1	1
151	1	1	1	1	1	1	1
152	1	1	1	1	1	1	1
153	1	1	1	1	1	1	1
154	1	1	1	1	1	1	1
155	0	1	1	1	1	1	1
156	0	1	0	1	0	1	1
157	0	1	0	0	0	0	1
158	0	1	0	0	0	0	0
159	0	1	0	0	0	0	0	1
160	1	1	0	0	0	0	0	1
161	1	1	0	0	0	0	0	1
162	1	1	0	0	0	0	0	1
163	1	1	0	0	0	0	0	1
164	1	1	0	0	0	0	0	1
165	0	1	0	0	0	0	0	1
166	0	1	1	0	0	0	0	1
167	0	1	1	1	0	0	0	1
168	0	1	1	1	0	0	0	1
169	0	1	1	1	0	0	0	1
170	1	1	1	1	0	0	0	1
171	1	1	1	1	0	0	0	1

Figure 3-11

172	1	1	1	1	0	0	0	1
173	1	1	1	1	0	0	0	1
174	1	1	1	1	0	0	0	1
175	0	1	1	1	0	0	0	1
176	0	1	0	1	1	0	0	1
177	0	1	0	0	1	0	0	1
178	0	1	0	0	1	0	0	1
179	0	1	0	0	1	0	0	1
180	1	1	0	0	1	0	0	1
181	1	1	0	0	1	0	0	1
182	1	1	0	0	1	0	0	1
183	1	1	0	0	1	0	0	1
184	1	1	0	0	1	0	0	1
185	0	1	0	0	1	0	0	1
186	0	1	1	0	1	0	0	1
187	0	1	1	1	1	0	0	1
188	0	1	1	1	1	0	0	1
189	0	1	1	1	1	0	0	1
190	1	1	1	1	1	0	0	1
191	1	1	1	1	1	0	0	1
192	1	1	1	1	1	0	0	1
193	1	1	1	1	1	0	0	1
194	1	1	1	1	1	0	0	1
195	0	1	1	1	1	0	0	1
196	0	1	0	1	0	0	0	1
197	0	1	0	0	0	1	0	1
198	0	1	0	0	0	1	0	1
199	0	1	0	0	0	1	0	1
200	1	1	0	0	0	1	0	1
201	1	1	0	0	0	1	0	1
202	1	1	0	0	0	1	0	1
203	1	1	0	0	0	1	0	1
204	1	1	0	0	0	1	0	1
205	0	1	0	0	0	1	0	1
206	0	1	1	0	0	1	0	1
207	0	1	1	1	0	1	0	1
208	0	1	1	1	0	1	0	1
209	0	1	1	1	0	1	0	1
210	1	1	1	1	0	1	0	1
211	1	1	1	1	0	1	0	1
212	1	1	1	1	0	1	0	1
213	1	1	1	1	0	1	0	1
214	1	1	1	1	0	1	0	1
215	0	1	1	1	0	1	0	1
216	0	1	0	1	1	1	0	1
217	0	1	0	0	1	1	0	1
218	0	1	0	0	1	1	0	1
219	0	1	0	0	1	1	0	1
220	1	1	0	0	1	1	0	1
221	1	1	0	0	1	1	0	1
222	1	1	0	0	1	1	0	1

Figure 3-12

223	1	1	0	0	1	1	0	1
224	1	1	0	0	1	1	0	1
225	0	1	0	0	1	1	0	1
226	0	1	1	0	1	1	0	1
227	0	1	1	1	1	1	0	1
228	0	1	1	1	1	1	0	1
229	0	1	1	1	1	1	0	1
230	1	1	1	1	1	1	0	1
231	1	1	1	1	1	1	0	1
232	1	1	1	1	1	1	0	1
233	1	1	1	1	1	1	0	1
234	1	1	1	1	1	1	0	1
235	0	1	1	1	1	1	0	1
236	0	1	0	1	0	1	0	1
237	0	1	0	0	0	0	0	1
238	0	1	0	0	0	0	1	1
239	0	1	0	0	0	0	1	1
240	1	1	0	0	0	0	1	1
241	1	1	0	0	0	0	1	1
242	1	1	0	0	0	0	1	1
243	1	1	0	0	0	0	1	1
244	1	1	0	0	0	0	1	1
245	0	1	0	0	0	0	1	1
246	0	1	1	0	0	0	1	1
247	0	1	1	1	0	0	1	1
248	0	1	1	1	0	0	1	1
249	0	1	1	1	0	0	1	1
250	1	1	1	1	0	0	1	1
251	1	1	1	1	0	0	1	1
252	1	1	1	1	0	0	1	1
253	1	1	1	1	0	0	1	1
254	1	1	1	1	0	0	1	1
255	0	1	1	1	0	0	1	1
256	0	1	0	1	1	0	1	1
257	0	1	0	0	1	0	1	1
258	0	1	0	0	1	0	1	1
259	0	1	0	0	1	0	1	1
260	1	1	0	0	1	0	1	1
261	1	1	0	0	1	0	1	1
262	1	1	0	0	1	0	1	1
263	1	1	0	0	1	0	1	1
264	1	1	0	0	1	0	1	1
265	0	1	0	0	1	0	1	1
266	0	1	1	0	1	0	1	1
267	0	1	1	1	1	0	1	1
268	0	1	1	1	1	0	1	1
269	0	1	1	1	1	0	1	1
270	1	1	1	1	1	0	1	1
271	1	1	1	1	1	0	1	1
272	1	1	1	1	1	0	1	1
273	1	1	1	1	1	0	1	1

Figure 3-13

274	1	1	1	1	1	0	1	1		
275	0	1	1	1	1	0	1	1		
276	0	1	0	1	0	0	1	1		
277	0	1	0	0	0	1	1	1		
278	0	1	0	0	0	1	1	1	0	
279	0	1	0	0	0	1	1	1	0	
280	1	1	0	0	0	1	1	1	0	
281	1	1	0	0	0	1	1	1	0	
282	1	1	0	0	0	1	1	1	0	
283	1	1	0	0	0	1	1	1	0	
284	1	1	0	0	0	1	1	1	0	
285	0	1	0	0	0	1	1	1	0	
286	0	1	1	0	0	1	1	1	0	
287	0	1	1	1	0	1	1	1	0	
288	0	1	1	1	0	1	1	1	0	
289	0	1	1	1	0	1	1	1	0	
290	1	1	1	1	0	1	1	1	0	
291	1	1	1	1	0	1	1	1	0	
292	1	1	1	1	0	1	1	1	0	
293	1	1	1	1	0	1	1	1	0	
294	1	1	1	1	0	1	1	1	0	
295	0	1	1	1	0	1	1	1	0	
296	0	1	0	1	1	1	1	1	0	
297	0	1	0	0	1	1	1	1	0	0
298	0	1	0	1	1	1	1	1	0	0
299	0	1	0	1	1	1	1	1	0	0
300	1	1	0	1	1	1	1	1	0	0
301	1	1	0	1	1	1	1	1	0	0
302	1	1	0	1	1	1	1	1	0	0
303	1	1	0	1	1	1	1	1	0	0
304	1	1	0	1	1	1	1	1	0	0
305	0	1	0	1	1	1	1	1	0	0
306	0	1	0	1	0	1	1	1	0	0
307	0	1	0	1	0	0	1	1	0	1
308	0	1	0	0	0	0	0	1	1	1
309	0	1	0	0	0	0	0	0	1	1
310	1	1	0	0	0	0	0	0	1	1
311	1	1	0	0	0	0	0	0	1	1
312	1	1	0	0	0	0	0	0	1	1
313	1	1	0	0	0	0	0	0	1	1
314	1	1	0	0	0	0	0	0	1	1
315	0	1	0	0	0	0	0	0	1	1
316	0	1	1	0	0	0	0	0	1	1
317	0	1	1	1	0	0	0	0	1	1
318	0	1	1	1	0	0	0	0	1	1
319	0	1	1	1	0	0	0	0	1	1
320	1	1	1	1	0	0	0	0	1	1
321	1	1	1	1	0	0	0	0	1	1
322	1	1	1	1	0	0	0	0	1	1
323	1	1	1	1	0	0	0	0	1	1
324	1	1	1	1	0	0	0	0	1	1

Figure 3-14

325	0	1	1	1	0	0	0	0	1	1
326	0	1	0	1	1	0	0	0	1	1
327	0	1	0	0	1	0	0	0	1	1
328	0	1	0	0	1	0	0	0	1	1
329	0	1	0	0	1	0	0	0	1	1
330	1	1	0	0	1	0	0	0	1	1
331	1	1	0	0	1	0	0	0	1	1
332	1	1	0	0	1	0	0	0	1	1
333	1	1	0	0	1	0	0	0	1	1
334	1	1	0	0	1	0	0	0	1	1
335	0	1	0	0	1	0	0	0	1	1
336	0	1	1	0	1	0	0	0	1	1
337	0	1	1	1	1	0	0	0	1	1
338	0	1	1	1	1	0	0	0	1	1
339	0	1	1	1	1	0	0	0	1	1
340	1	1	1	1	1	0	0	0	1	1
341	1	1	1	1	1	0	0	0	1	1
342	1	1	1	1	1	0	0	0	1	1
343	1	1	1	1	1	0	0	0	1	1
344	1	1	1	1	1	0	0	0	1	1
345	0	1	1	1	1	0	0	0	1	1
346	0	1	0	1	0	0	0	0	1	1
347	0	1	0	0	0	1	0	0	1	1
348	0	1	0	0	0	1	0	0	1	1
349	0	1	0	0	0	1	0	0	1	1
350	1	1	0	0	0	1	0	0	1	1

PRINT HAZARDS

TYPE1 HAZARD ON SIGNAL NCCNCT AT TIME 298. PULSE VALUE WAS 0

END OF STEP * SIMULATE PFAMD

Figure 3-15

```
* EXIT

DLL5 EXECUTION COMPLETED
```

Figure 3-16

* SIMULATE (3.2.3)
* CHANGE (3.2.4)
* OUTPUT (3.2.5)
* EXIT (3.2.6)

3.2.1 OPEN

This instruction has the form,

* OPEN DATAFILE filename

Its function is twofold; it establishes the existence of a working file, and it replaces the ADD instruction.

The data file specified by the variable "filename" becomes the working file, replacing any previously named working file; if there is no file which has this name, one is created.

In the example, Fig. 3-1, "CREATE FILE" would now be replaced by "OPEN DATAFILE".

If an OPEN instruction is followed by data cards, they are assumed to be of the form described for the ADD instruction described in E-2265 (Section 3.5.5). "OPEN DATAFILE" replaces "GET" in Fig. 3-5. The first device being added to the file is a NAND gate (which was defined in Fig. 3-2) with identification "1", and it is placed on drawing "1" at coordinates (1, 1), its output pin A is the source of signal "RSTNCT/", and its input pin B is connected to "RSTNCT".

3.2.2 DEFINE

The DEFINE instruction (See E-2265, 3.2, 3.3.1) is in working condition. It has the form,

* DEFINE type

where the variable "type" is the name given to the device which is defined by the subsequent data cards. The new device definition is added to the glossary of the working file.

Figures 3-2 through 3-5 show three devices being defined, a 4-input NAND gate, a J-K flip-flop, and an oscillator.

3.2.3 SIMULATE

The SIMULATE instruction has the form,

* SIMULATE (filename)

The parentheses indicate that the variable "filename" is optional. If present, "filename" is the name of the (read) data file to be simulated; if missing, the file simulated is the current working file.

The following subinstructions are available:

EVENT	(3.3.2,b; 3.3.3,a41)**
SEQUENCE	(3.3.2,b)
TRACE	(3.3.2,c1)
SAMPLE	(3.3.2,c2; 3.3.3,a4a)
RUN	(3.3.2,d; 3.3.3,a4b)
INITIAL0	(3.3.3,c)
INITIAL1	(3.3.3,c)
PROP	(formerly PROPAGATE: (3.3.3,c)
MINPULSE	
PRINT	

The MINPULSE instruction is used to define the minimum number of time units a signal must either be 0 or 1 for a hazard not to exist.

A sample simulation is shown in Figs. 3-6 through 3-15. The PRINT instruction is used in Figs. 3-6 and 3-7 to obtain the lists shown, and in Fig. 3-8 to obtain the linear trace. Thirteen other PRINT options are available at this time.

3.2.4 CHANGE

The operations which can now be performed with the CHANGE instruction,

```
* CHANGE
```

are (1) the identification of a drawing, signal, or device may be changed, and (2) a device may be moved to a different drawing or to a different location on the same drawing. Other functions have been coded, but not debugged.

3.2.5 OUTPUT

This instruction,

```
* OUTPUT (filename)
```

is used to generate signal lists and other useful data from either the file "filename" or, if this variable is missing, from the current working file.

3.2.6 EXIT

An EXIT card is needed at the end of each OLLS run to insure a normal FILLIP termination. Its format is simply,

```
* EXIT
```

**Numbers in parentheses are references to sections of Report E-2265.

3.3 Contemplated Programs

The following asterisk instructions are planned for future implementation:

DELETE	(3.3.1)
DELETE TYPE	(3.3.2)
COPY	(3.3.3)
ASSEMBLE	(3.3.4)

3.3.1 DELETE

The function of the asterisk card

* DELETE

is described in the report (Section 3.5.9)

3.3.2 DELETE TYPE

The instruction

* DELETE TYPE type

will be used to remove the definition named "type" from the glossary of the working file (See Section 3.5.10 of the report).

3.3.3 COPY

The COPY instruction, which will have the format

* COPY FROM filename

will be used to duplicate into the working file useful device definitions or entire drawings. The source of these definitions and drawings is the file "filename".

3.3.4 ASSEMBLE

The ASSEMBLE instruction,

* ASSEMBLE type

will be used to create a device definition named "type" in the glossary of the working file; data following the asterisk card indicates several instances to be combined to form this definition. (See Section 3.2.4 of the report).

4. THE CRT INTERACTIVE SYSTEM

This section was called Section 3.6 in E-2265 but because the CRT programs now operate as a stand-alone system and because it is this system which we feel has the most promise for the future, we present here a new section with the included figures which illustrate our progress to date.

The CRT system is written in IBM machine language. A data structure very similar to that used by the FILLIP system (See E-2265 Section 3) was implemented by machine language subroutines which allow the program to operate on a file which is much larger than the available CPU core capacity. These routines can rapidly follow pointers to data which is not currently in core if so required by the operator at the CRT.

As described in E-2265 our design philosophy for the CRT system has been to relieve the logic designer of the burden of learning detailed card formats and conventions and in general to show him the way as much as possible. Instructions to the designer appear prominently on the screen when appropriate. Options which are logically available for him to select at any time are indicated by a "#" to the left of the option. We call this character a "light button", and, to select it, the designer merely points the light pen at it and depresses the tip switch in the pen. When the designer has selected a particular option, he is given feedback from the program by changing the "#" to the character "X" to indicate selected. This not only reminds him of what option he is currently operating, but it should help prevent him from "fat fingering" the light buttons accidentally with the light pen (which is somewhat similar to a blunderbuss).

We have tried to avoid use of the alphameric keyboard except where absolutely necessary, i.e., when creating new names for files, devices, signals, etc. Again this is to relieve the designer of any opportunity to do any thinking except on his design problem. We have provided a parallel set of light buttons to those on the screen with the programmed function keyboard (PFK). We found that after a designer has had some hours of experience operating the CRT system, he can work slightly faster by using the PFK if he learns by memory the assignments

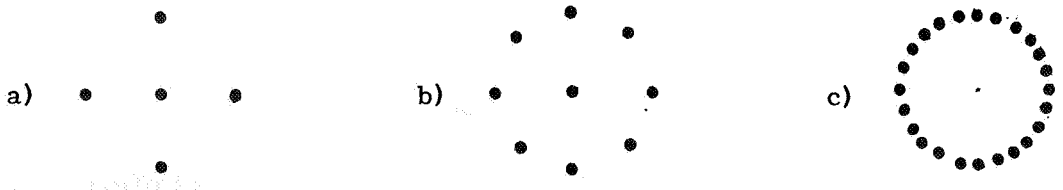
of the buttons. One important factor which we found essential to consider is the time and trouble required to hang up the light pen in order to type a character or press a button. This may seem trivial, but experience has shown us the opposite.

4.1 Light Pen Tracking

One area where both "light-buttons" and the keyboard are inadequate is in describing lines and shapes to the logic file. Since the light pen is only a light detector, the only way in which the program can know the X, Y coordinates of the pen on the screen at any time is for the pen to detect light from some symbol which is already on the screen at coordinates known to the program. The program can then take logical action to move the symbol to a new set of coordinates and add the new X, Y point to the line segment currently being drawn. Basically the program must display a tracking symbol which traps the light pen within its light boundaries. Thus whenever the designer moves the light pen, the program receives an interrupt from the light pen detect and moves the tracking symbol in the direction of the detect.

Although there are many possibilities in the design of tracking symbols such as the static display of a circle, square, cross, spiral, etc., or a dynamic display of a random pattern of points or a regularly scanned area of the screen, only a few are well suited to a particular application. Dynamic patterns require local hardware able to update the display continuously and handle a very high interrupt rate.

Static patterns must be of just the right shape, size, light intensity, and light sensitivity to provide the user with sufficient degrees of freedom to accomplish his task. In the three tracking symbols shown below, we have combined the characteristics of our available hardware with the requirements of OLLS to produce a very flexible facility for light pen tracking. (Each symbol is about 1/2 inch in diameter.)



Symbol a) encourages long straight lines since the pen can detect light only from a point which is vertically or horizontally related to the center. Symbol b) can be used to draw lines at 45^{o} as well as vertically or horizontally. Symbol c) has enough points around the center so that smooth curves can be drawn.

The basic symbols above have been augmented by providing the designer with some frills which are available for him to select such as tracking magnification for very detailed work, a means of moving the symbol unattached from the line segment he is drawing, and a means of turning the symbol on and off while tracking. (Remember the pen is trapped by the symbol and if the symbol could not be turned off, the designer could not even pick up the pen without leaving a light smear on his work.) The point in the center of each symbol is used as an alternate action switch which causes the program to sensitize (enable) or desensitize (disable) the points around the center for light per detects. We have found that with these frills, light pen tracking can be very easy and effective. (See Fig. 4-25)

Two things are essential to economical light pen tracking. One is software designed with the user in mind and the other is hardware which allows interrupts to be processed rapidly without tying up too much CPU. For a time sharing environment this would be impossible without a small dedicated computer to buffer interrupts and update the display. In a multiprogrammed environment such as ours, interrupts can be handled more readily by the CPU since the programs to handle them are in core when needed. In either environment a display unit such as the IBM 2250 Model 3 which can execute buffer subroutines would be better than the Model 1. The buffer subroutine can update the position of the tracking symbol and add points to the segment being drawn interrupting the CPU only when the operator wishes to transmit a complete line segment.

The following figures are reproductions of photographs taken at the IBM 2250 CRT console as the designer uses almost as much of the system as is operational to date. As of this writing two important sets of menus, DRAWING MANIPULATE and OUTPUT OPTIONS, are in the final stages of debugging.

Figure 4-1

When the system first comes on, we see the major options available and that we have neither a Read Only File nor a Working File.

The designer selects FILE MANIPULATE by touching "#" with the light pen.

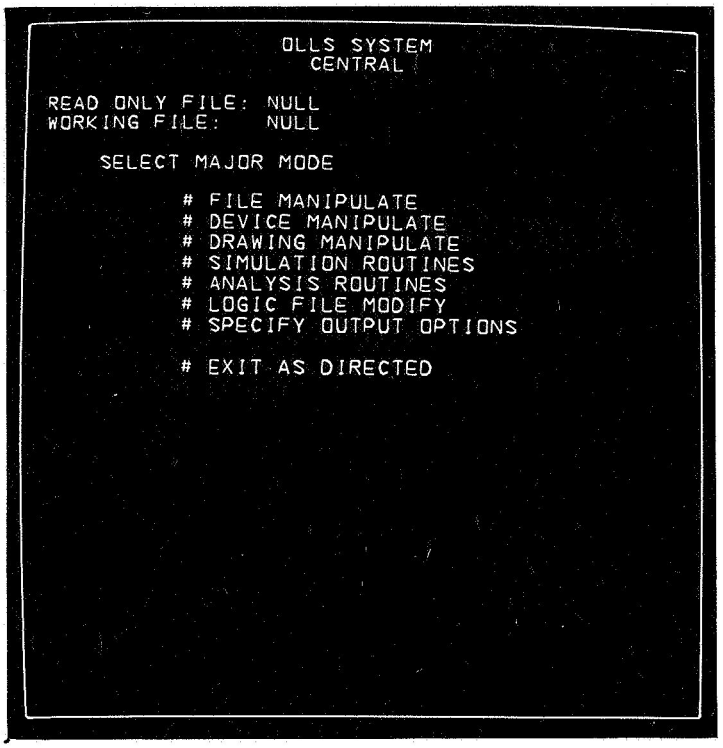

OLLS SYSTEM
CENTRAL
READ ONLY FILE: NULL
WORKING FILE: NULL
SELECT MAJOR MODE
FILE MANIPULATE
DEVICE MANIPULATE
DRAWING MANIPULATE
SIMULATION ROUTINES
ANALYSIS ROUTINES
LOGIC FILE MODIFY
SPECIFY OUTPUT OPTIONS
EXIT AS DIRECTED

Figure 4-2

The File Manipulate menu shows the list of available files and the options available.

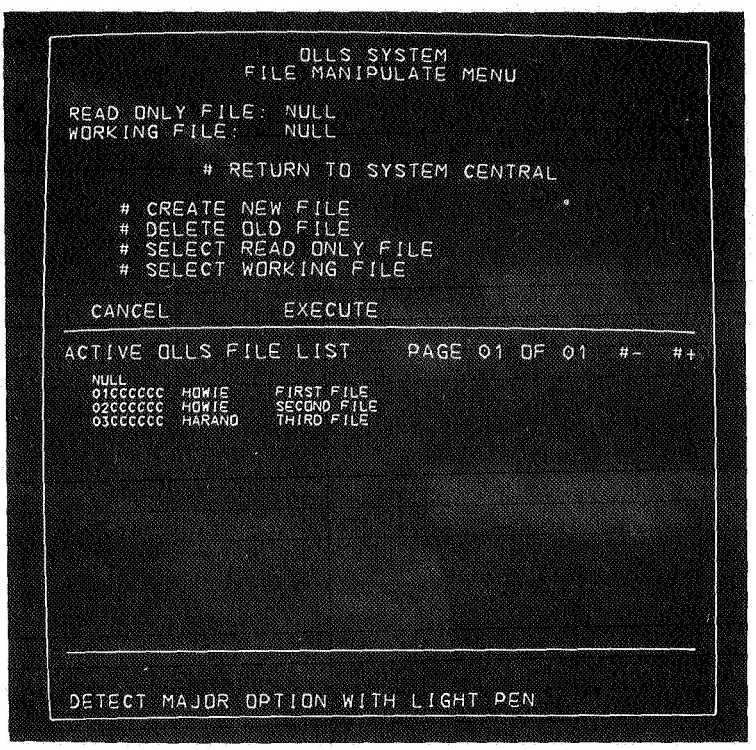
Directions or messages to the designer appear on the screen (usually at the bottom) when appropriate.

Figure 4-3

The designer detects

SELECT WORKING FILE.

An "X" always appears in place of the "#" which the designer detected. This shows him positively what he is currently doing lest he forget or blunder with the light pen.


OLLS SYSTEM
FILE MANIPULATE MENU
READ ONLY FILE: NULL
WORKING FILE: NULL
RETURN TO SYSTEM CENTRAL
CREATE NEW FILE
DELETE OLD FILE
SELECT READ ONLY FILE
SELECT WORKING FILE
CANCEL EXECUTE
ACTIVE OLLS FILE LIST PAGE 01 OF 01 #- #+
NULL
01CCCCCC HOWIE FIRST FILE
02CCCCCC HOWIE SECOND FILE
03CCCCCC HARAND THIRD FILE
DETECT MAJOR OPTION WITH LIGHT PEN

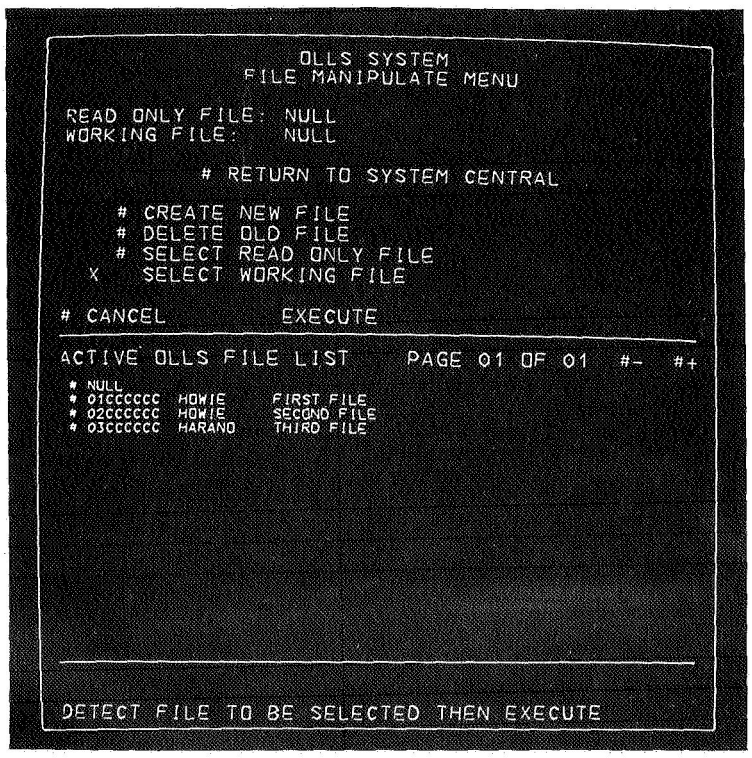

OLLS SYSTEM
FILE MANIPULATE MENU
READ ONLY FILE: NULL
WORKING FILE: NULL
RETURN TO SYSTEM CENTRAL
CREATE NEW FILE
DELETE OLD FILE
SELECT READ ONLY FILE
X SELECT WORKING FILE
CANCEL EXECUTE
ACTIVE OLLS FILE LIST PAGE 01 OF 01 #- #+
* NULL
* 01CCCCCC HOWIE FIRST FILE
* 02CCCCCC HOWIE SECOND FILE
* 03CCCCCC HARAND THIRD FILE
DETECT FILE TO BE SELECTED THEN EXECUTE

Figure 4-4

The designer detects

03CCCCCC to be his working file. This file was created by

CREATE NEW FILE on an earlier run.

Figure 4-5

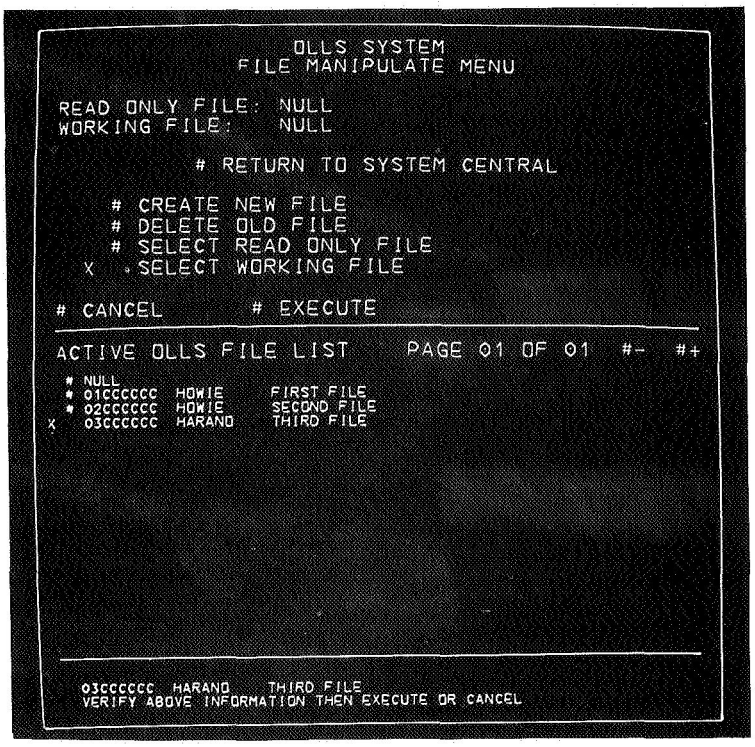
After the designer detects

EXECUTE

we see that the WORKING FILE is

03CCCCCC as requested. (In fact the entire file is moved from the IBM 2314 disk pack where all the files are stored to the IBM 2301 drum. The drum operates much faster than disk and protects the old copy of the file in case of some system disaster.)

NOTE: The actual selection of which storage devices will be allocated to the various files is determined at execution time by the Job Control Language statements which invoke OLLS. The JCL can also select a different "ACTIVE OLLS FILE LIST" for different runs thus providing complete flexibility to run OLLS in any IBM 360 environment or to transport an OLLS file or the OLLS System to a different computing facility.


OLLS SYSTEM
FILE MANIPULATE MENU
READ ONLY FILE: NULL
WORKING FILE: NULL
RETURN TO SYSTEM CENTRAL
CREATE NEW FILE
DELETE OLD FILE
SELECT READ ONLY FILE
X # SELECT WORKING FILE
CANCEL # EXECUTE
ACTIVE OLLS FILE LIST PAGE 01 OF 01 #- #+
NULL
01CCCCCC HOWIE FIRST FILE
02CCCCCC HOWIE SECOND FILE
X 03CCCCCC HARAND THIRD FILE
03CCCCCC HARAND THIRD FILE
VERIFY ABOVE INFORMATION THEN EXECUTE OR CANCEL

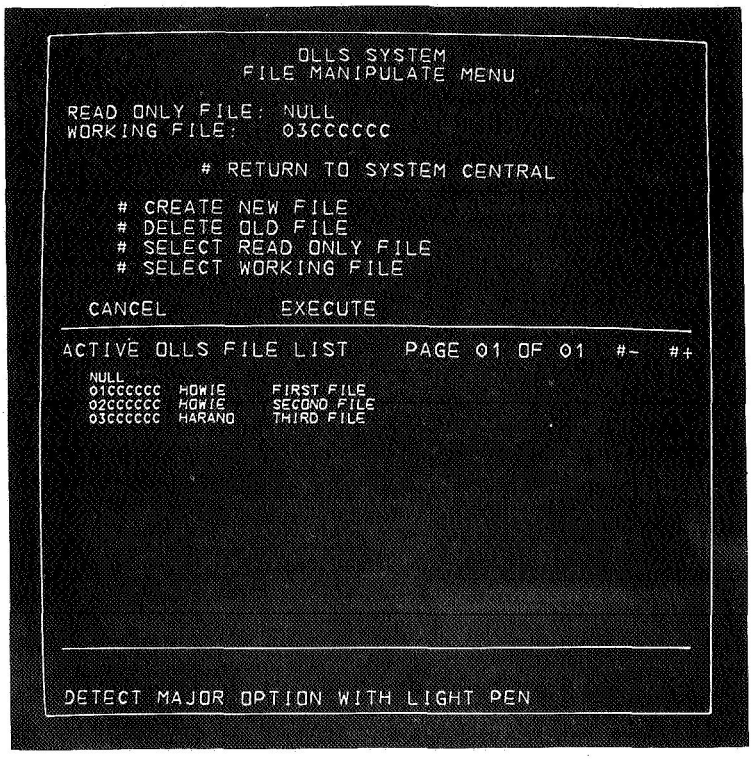

OLLS SYSTEM
FILE MANIPULATE MENU
READ ONLY FILE: NULL
WORKING FILE: 03CCCCCC
RETURN TO SYSTEM CENTRAL
CREATE NEW FILE
DELETE OLD FILE
SELECT READ ONLY FILE
SELECT WORKING FILE
CANCEL EXECUTE
ACTIVE OLLS FILE LIST PAGE 01 OF 01 #- #+
NULL
01CCCCCC HOWIE FIRST FILE
02CCCCCC HOWIE SECOND FILE
03CCCCCC HARAND THIRD FILE
DETECT MAJOR OPTION WITH LIGHT PEN

Figure 4-6

The designer has detected

```
# SELECT READ ONLY FILE
# 02CCCCCC                and
# EXECUTE
```

to check out a read only file.

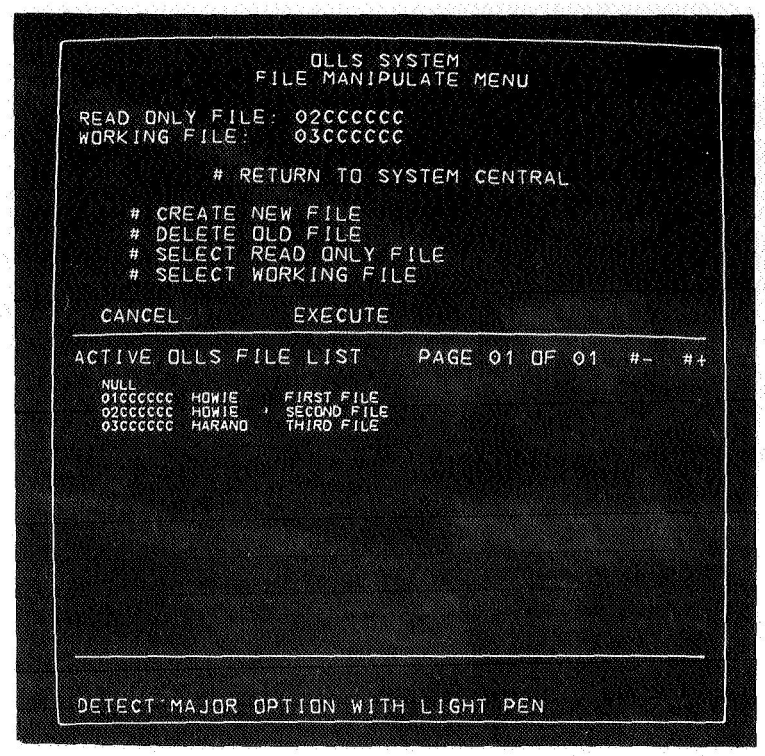
Since he is now finished with this menu he detects

```
# RETURN TO SYSTEM CENTRAL
```

Figure 4-7

Again we see the major options available, but now we have an active working file and can set out to do some useful work:

```
# DEVICE MANIPULATE
```


OLLS SYSTEM
FILE MANIPULATE MENU
READ ONLY FILE: 02CCCCCC
WORKING FILE: 03CCCCCC
RETURN TO SYSTEM CENTRAL
CREATE NEW FILE
DELETE OLD FILE
SELECT READ ONLY FILE
SELECT WORKING FILE
CANCEL
EXECUTE
ACTIVE OLLS FILE LIST
PAGE 01 OF 01
#-
#+
NULL
01CCCCCC HOWIE FIRST FILE
02CCCCCC HOWIE SECOND FILE
03CCCCCC HARAND THIRD FILE
DETECT MAJOR OPTION WITH LIGHT PEN

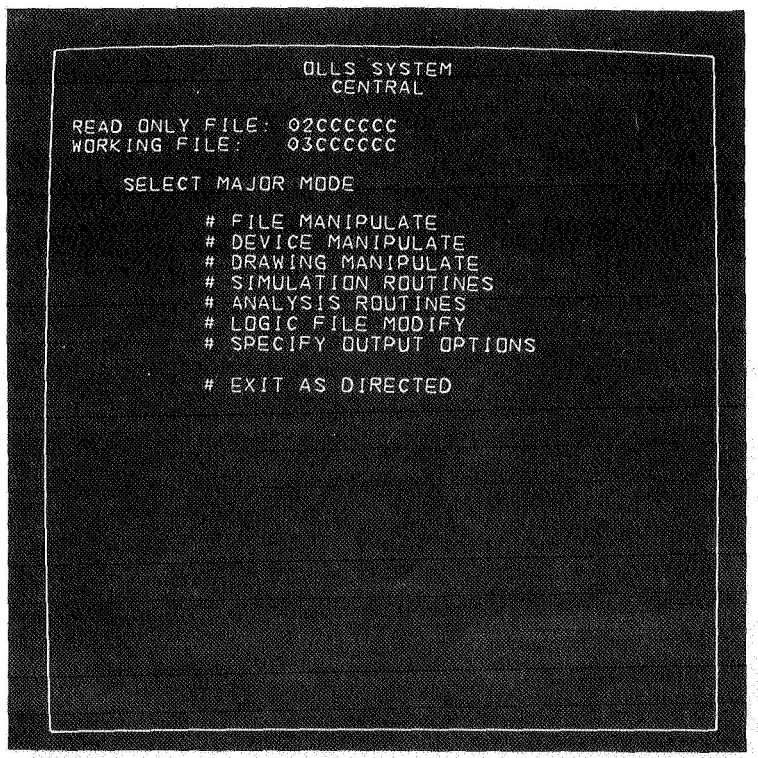

OLLS SYSTEM
CENTRAL
READ ONLY FILE: 02CCCCCC
WORKING FILE: 03CCCCCC
SELECT MAJOR MODE
FILE MANIPULATE
DEVICE MANIPULATE
DRAWING MANIPULATE
SIMULATION ROUTINES
ANALYSIS ROUTINES
LOGIC FILE MODIFY
SPECIFY OUTPUT OPTIONS
EXIT AS DIRECTED

Figure 4-8

The DEVICE CENTRAL menu shows the list of available devices in the designer's working file (03CCCCCC). These devices are his to copy, delete, or modify (at his own peril) to suit his design needs. It is from this list that he chooses devices for his drawings.

Figure 4-9

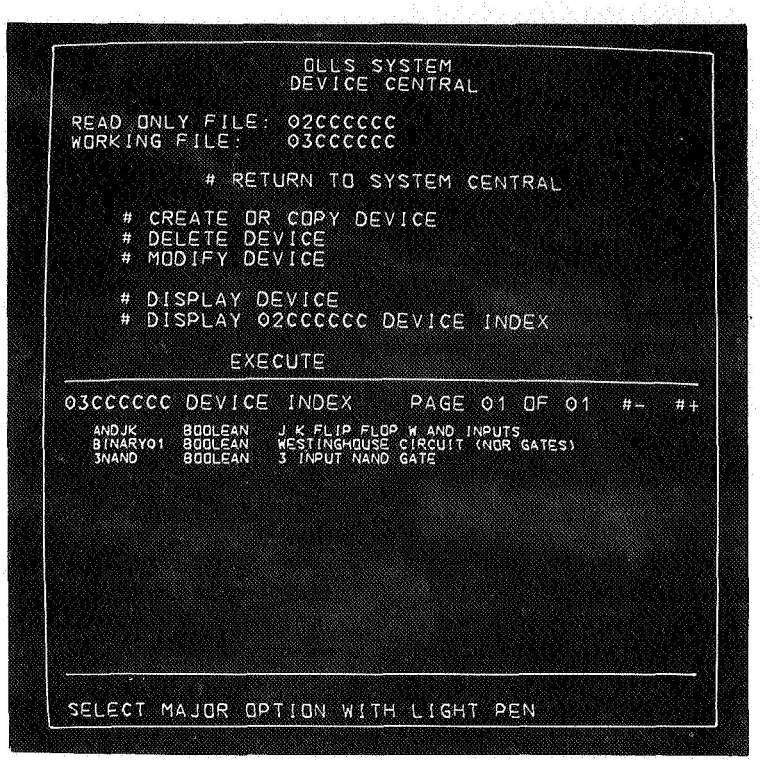
The designer has detected

```
# DISPLAY 02CCCCCC DEVICE INDEX
```

then

```
# EXECUTE
```

to display the list of devices in the read only file (02CCCCCC). These devices are available only for the designer to display or to copy to save him the time and trouble of redefining a useful device himself.


OLLS SYSTEM
DEVICE CENTRAL
READ ONLY FILE: 02CCCCCC
WORKING FILE: 03CCCCCC
RETURN TO SYSTEM CENTRAL
CREATE OR COPY DEVICE
DELETE DEVICE
MODIFY DEVICE
DISPLAY DEVICE
DISPLAY 02CCCCCC DEVICE INDEX
EXECUTE
03CCCCCC DEVICE INDEX PAGE 01 OF 01 #- #+
ANDJK BOOLEAN J K FLIP FLOP W AND INPUTS
BINARY01 BOOLEAN WESTINGHOUSE CIRCUIT (NOR GATES)
3NAND BOOLEAN 3 INPUT NAND GATE
SELECT MAJOR OPTION WITH LIGHT PEN

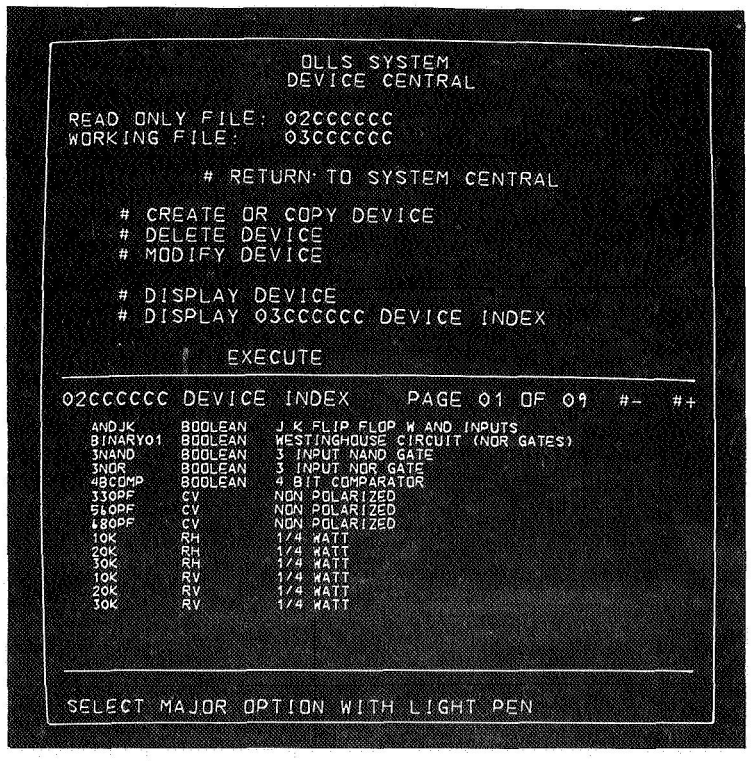

OLLS SYSTEM
DEVICE CENTRAL
READ ONLY FILE: 02CCCCCC
WORKING FILE: 03CCCCCC
RETURN TO SYSTEM CENTRAL
CREATE OR COPY DEVICE
DELETE DEVICE
MODIFY DEVICE
DISPLAY DEVICE
DISPLAY 03CCCCCC DEVICE INDEX
EXECUTE
02CCCCCC DEVICE INDEX PAGE 01 OF 09 #- #+
ANDJK BOOLEAN J K FLIP FLOP W AND INPUTS
BINARY01 BOOLEAN WESTINGHOUSE CIRCUIT (NOR GATES)
3NAND BOOLEAN 3 INPUT NAND GATE
3NOR BOOLEAN 3 INPUT NOR GATE
4BCOMP BOOLEAN 4 BIT COMPARATOR
330PF CV NON POLARIZED
560PF CV NON POLARIZED
680PF CV NON POLARIZED
10K RH 1/4 WATT
20K RH 1/4 WATT
30K RH 1/4 WATT
10K RV 1/4 WATT
20K RV 1/4 WATT
30k RV 1/4 WATT
SELECT MAJOR OPTION WITH LIGHT PEN

Figure 4-10

The designer wishes to examine a particular device (4BCOMP) in the read only file perhaps to pirate a good idea or to decide if he wants to copy it to this working file.

He detects

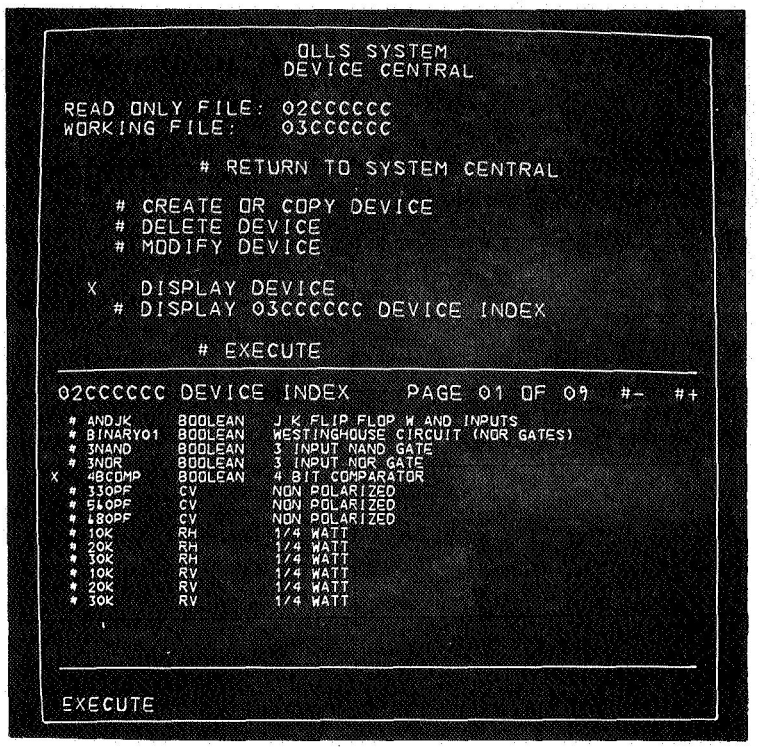
```
# DISPLAY DEVICE
# 4BCOMP
```

followed by

```
# EXECUTE
```

Figure 4-11

The DISPLAY DEVICE menu allows the designer to examine the device shape, terminals or equations. (He is now looking at shape and terminals.)


OLLS SYSTEM
DEVICE CENTRAL
READ ONLY FILE: 02CCCCCC
WORKING FILE: 03CCCCCC
RETURN TO SYSTEM CENTRAL
CREATE OR COPY DEVICE
DELETE DEVICE
MODIFY DEVICE
X DISPLAY DEVICE
DISPLAY 03CCCCCC DEVICE INDEX
EXECUTE
02CCCCCC DEVICE INDEX PAGE 01 OF 09 #- #+
ANDJK BOOLEAN J K FLIP FLOP W AND INPUTS
BINARY01 BOOLEAN WESTINGHOUSE CIRCUIT (NOR GATES)
3NAND BOOLEAN 3 INPUT NAND GATE
3NOR BOOLEAN 3 INPUT NOR GATE
X 4BCOMP BOOLEAN 4 BIT COMPARATOR
330PF CV NON POLARIZED
560PF CV NON POLARIZED
680PF CV NON POLARIZED
10K RH 1/4 WATT
20K RH 1/4 WATT
30K RH 1/4 WATT
10K RV 1/4 WATT
20K RV 1/4 WATT
30K RV 1/4 WATT
EXECUTE

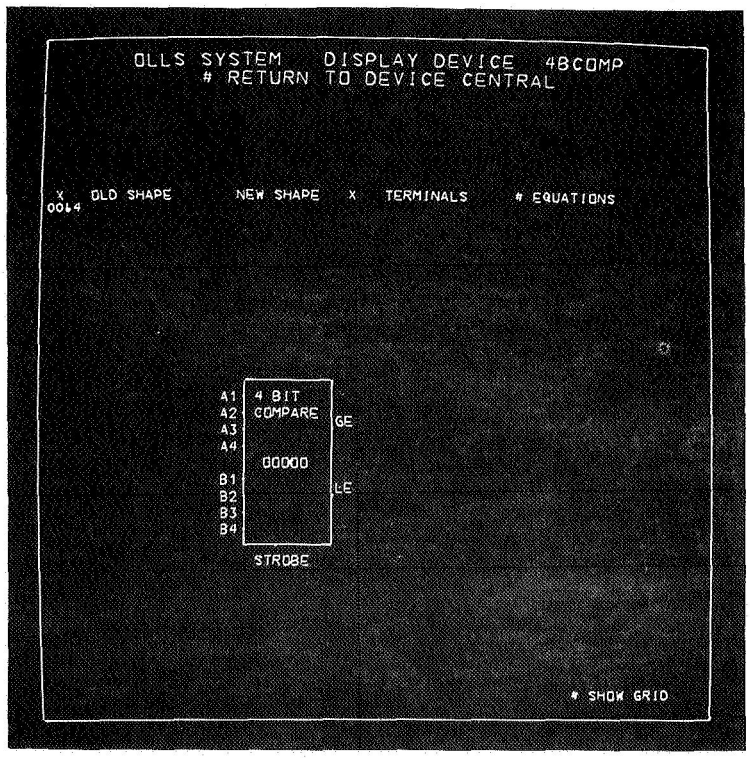

OLLS SYSTEM DISPLAY DEVICE 4BCOMP
RETURN TO DEVICE CENTRAL
X OLD SHAPE NEW SHAPE X TERMINALS # EQUATIONS
0014
A1
A2
A3
A4
4 BIT
COMPARE
GE
00000
B1
B2
B3
B4
LE
STROBE
SHOW GRID

Figure 4-12

The designer has detected

EQUATIONS

(and has turned off the terminals display for some reason by detecting

X ..... TERMINALS)

The equation list shows the two output equations and lists the inputs.

RETURN TO DEVICE CENTRAL

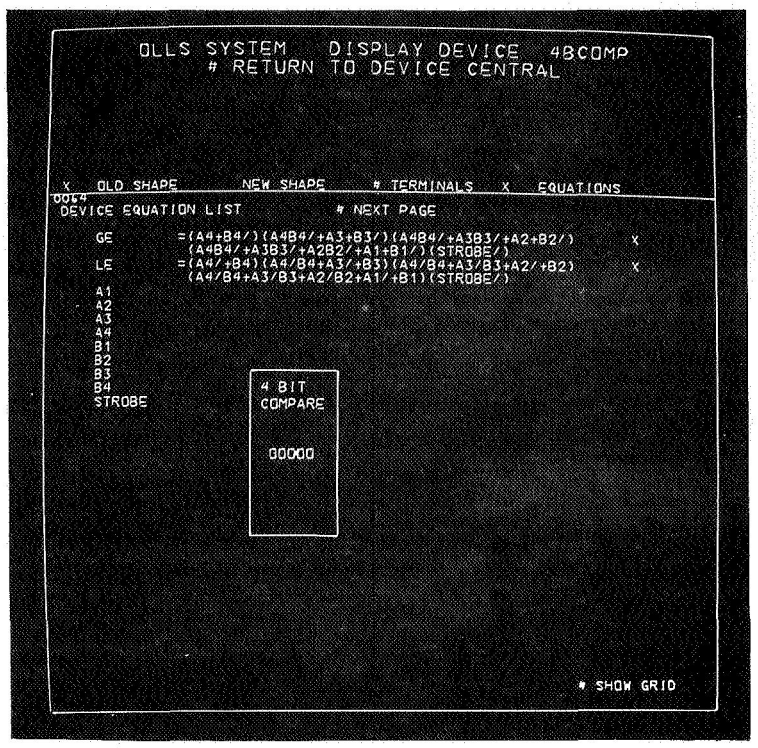
Figure 4-13

The designer wishes to display yet another device so he detects

DISPLAY DEVICE

BINARY01

EXECUTE


OLLS SYSTEM DISPLAY DEVICE 4BCOMP
RETURN TO DEVICE CENTRAL
X OLD SHAPE NEW SHAPE # TERMINALS X EQUATIONS
0064
DEVICE EQUATION LIST # NEXT PAGE
GE =(A4+B4/)(A4B4/+A3+B3/)(A4B4/+A3B3/+A2+B2/) X
(A4B4/+A3B3/+A2B2/+A1+B1/)(STROBE/)
LE =(A4/+B4)(A4/B4+A3/+B3)(A4/B4+A3/B3+A2/+B2) X
(A4/B4+A3/B3+A2/B2+A1/+B1)(STROBE/)
A1
A2
A3
A4
B1
B2
B3
B4
STROBE
4 BIT
COMPARE
00000
SHOW GRID

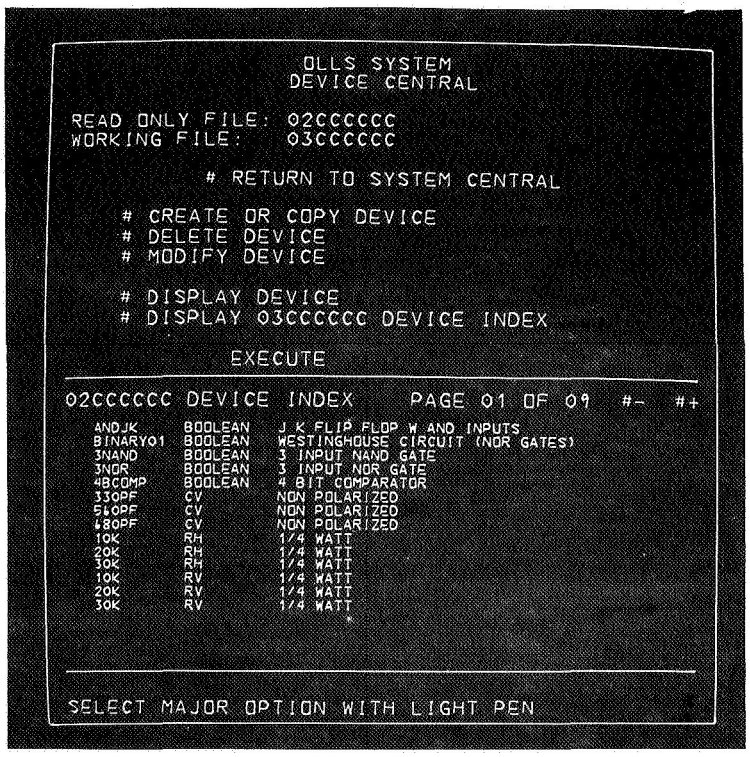

OLLS SYSTEM
DEVICE CENTRAL
READ ONLY FILE: 02CCCCCC
WORKING FILE: 03CCCCCC
RETURN TO SYSTEM CENTRAL
CREATE OR COPY DEVICE
DELETE DEVICE
MODIFY DEVICE
DISPLAY DEVICE
DISPLAY 03CCCCCC DEVICE INDEX
EXECUTE
02CCCCCC DEVICE INDEX PAGE 01 OF 09 #- #+
ANDJK BOOLEAN J K FLIP FLOP W AND INPUTS
BINARY01 BOOLEAN WESTINGHOUSE CIRCUIT (NOR GATES)
3NAND BOOLEAN 3 INPUT NAND GATE
3NOR BOOLEAN 3 INPUT NOR GATE
4BCOMP BOOLEAN 4 BIT COMPARATOR
330PF CV NON POLARIZED
560PF CV NON POLARIZED
680PF CV NON POLARIZED
10K RH 1/4 WATT
20K RH 1/4 WATT
30K RH 1/4 WATT
10K RV 1/4 WATT
20K RV 1/4 WATT
30K RV 1/4 WATT
SELECT MAJOR OPTION WITH LIGHT PEN

Figure 4-14

This shows an example of what we consider about the limit of complexity of a single component device. A more complicated device would better be defined as a "drawing" in its own right and built up from smaller component devices.

Figure 4-15

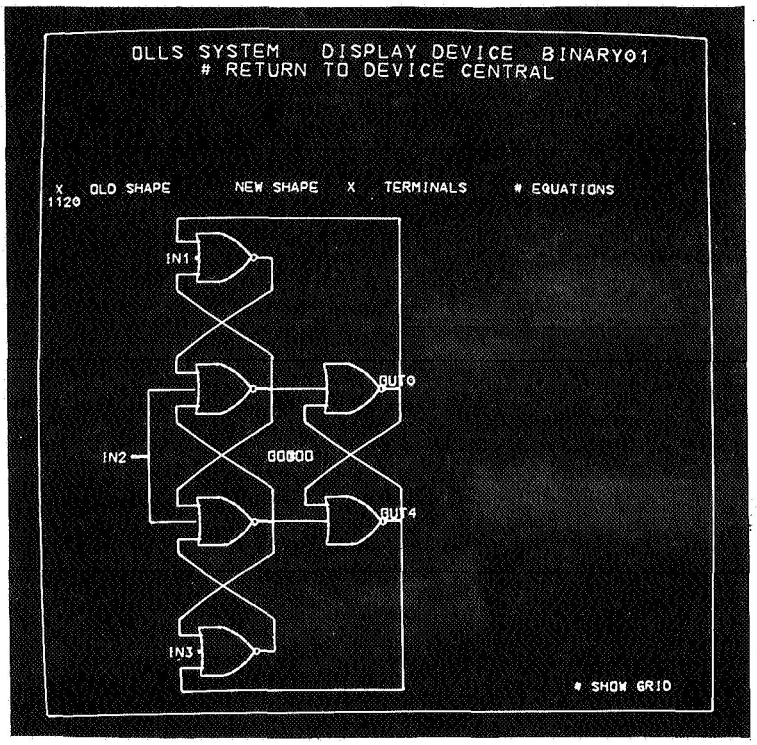
To create a new device he detects

CREATE OR COPY DEVICE

and begins typing in:

FLOP01 BOOLEAN FLIP FLOP (3NAND GATES)

This new device will be added to his working file when he presses the END KEY after typing.


OLLS SYSTEM DISPLAY DEVICE BINARY01
RETURN TO DEVICE CENTRAL
X OLD SHAPE
1120
NEW SHAPE
X TERMINALS
EQUATIONS
IN1
IN2
IN3
OUT0
OUT4
SHOW GRID

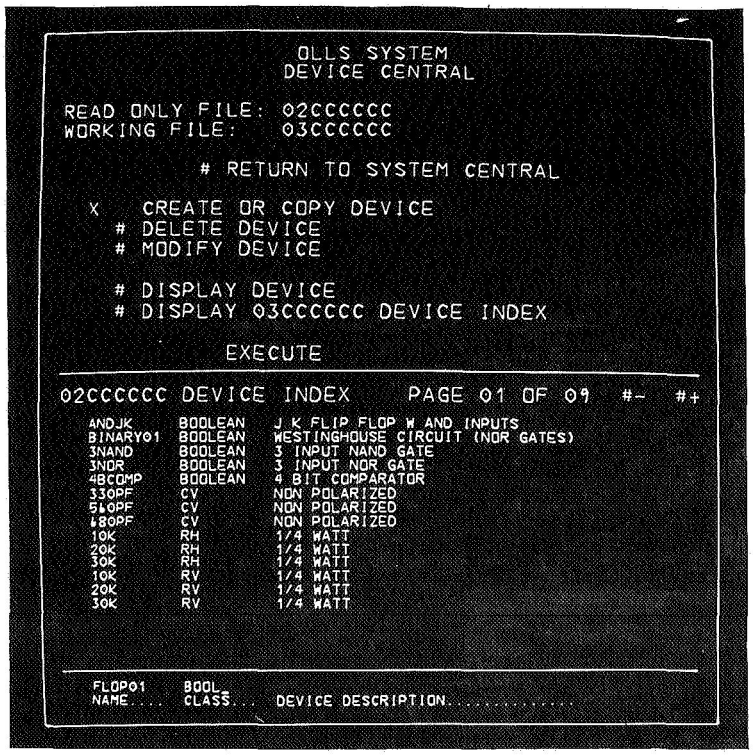

OLLS SYSTEM
DEVICE CENTRAL
READ ONLY FILE: 02CCCCCC
WORKING FILE: 03CCCCCC
RETURN TO SYSTEM CENTRAL
X CREATE OR COPY DEVICE
DELETE DEVICE
MODIFY DEVICE
DISPLAY DEVICE
DISPLAY 03CCCCCC DEVICE INDEX
EXECUTE
02CCCCCC DEVICE INDEX PAGE 01 OF 09 #- #+
ANDJK BOOLEAN J K FLIP FLOP W AND INPUTS
BINARY01 BOOLEAN WESTINGHOUSE CIRCUIT (NOR GATES)
3NAND BOOLEAN 3 INPUT NAND GATE
3NOR BOOLEAN 3 INPUT NOR GATE
4BCOMP BOOLEAN 4 BIT COMPARATOR
330PF CV NON POLARIZED
510PF CV NON POLARIZED
680PF CV NON POLARIZED
10K RH 1/4 WATT
20K RH 1/4 WATT
30K RH 1/4 WATT
10K RV 1/4 WATT
20K RV 1/4 WATT
30K RV 1/4 WATT
FLOP01 BOOL
NAME... CLASS... DEVICE DESCRIPTION.....

Figure 4-16

If the designer wishes his new device to be initially a copy of some other device, he detects that device, i.e.

3NAND BOOLEAN 3 INPUT NAND GATE

in this case before he detects

EXECUTE

(A more appropriate message at the bottom of the screen has been incorporated since the photograph.)

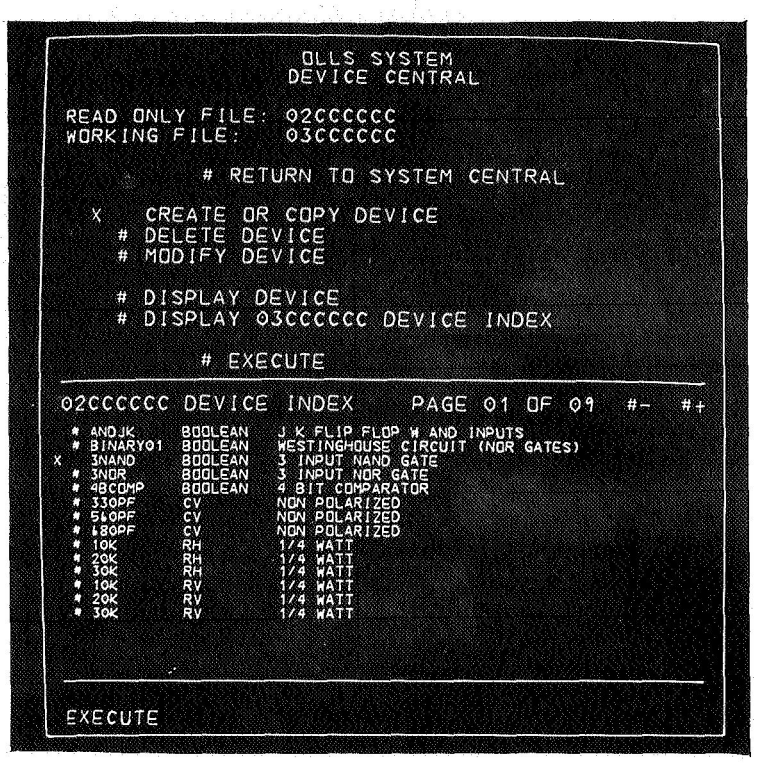
Figure 4-17

When the designer detects

DISPLAY 03CCCCCC DEVICE INDEX
(his working file)

EXECUTE,

he sees that FLOP01 has indeed been added to the list. It is identical to the 3NAND he copied. Had he not elected to copy another device, a so-called NULL device would have been created.


OLLS SYSTEM
DEVICE CENTRAL
READ ONLY FILE: 02CCCCCC
WORKING FILE: 03CCCCCC
RETURN TO SYSTEM CENTRAL
X CREATE OR COPY DEVICE
DELETE DEVICE
MODIFY DEVICE
DISPLAY DEVICE
DISPLAY 03CCCCCC DEVICE INDEX
EXECUTE
02CCCCCC DEVICE INDEX PAGE 01 OF 09 #- #+
ANDJK BOOLEAN J K FLIP FLOP W AND INPUTS
BINARY01 BOOLEAN WESTINGHOUSE CIRCUIT (NOR GATES)
X 3NAND BOOLEAN 3 INPUT NAND GATE
3NOR BOOLEAN 3 INPUT NOR GATE
4BCOMP BOOLEAN 4 BIT COMPARATOR
330PF CV NON POLARIZED
560PF CV NON POLARIZED
680PF CV NON POLARIZED
10K RH 1/4 WATT
20K RH 1/4 WATT
30K RH 1/4 WATT
10K RV 1/4 WATT
20K RV 1/4 WATT
30K RV 1/4 WATT
EXECUTE

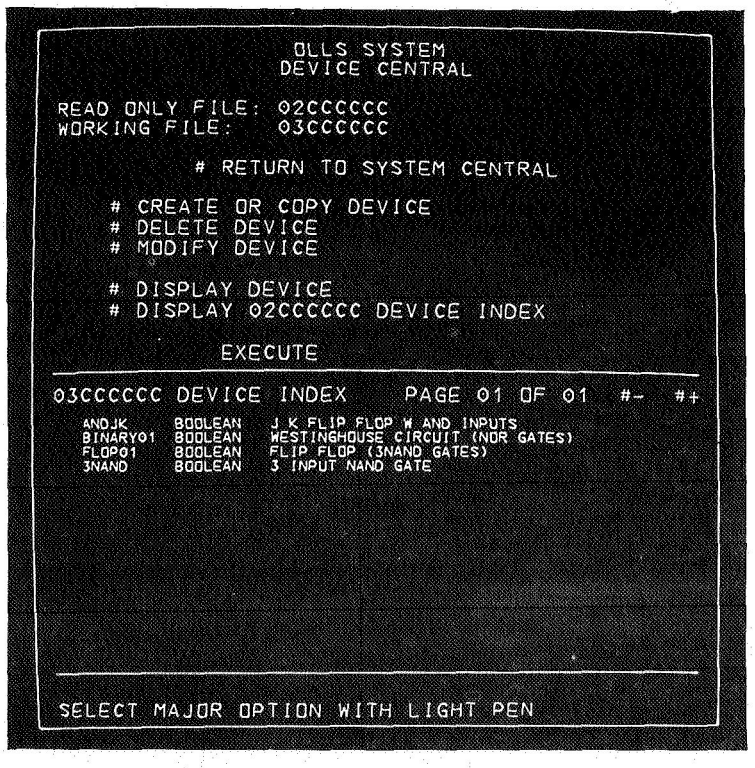

OLLS SYSTEM
DEVICE CENTRAL
READ ONLY FILE: 02CCCCCC
WORKING FILE: 03CCCCCC
RETURN TO SYSTEM CENTRAL
CREATE OR COPY DEVICE
DELETE DEVICE
MODIFY DEVICE
DISPLAY DEVICE
DISPLAY 02CCCCCC DEVICE INDEX
EXECUTE
03CCCCCC DEVICE INDEX PAGE 01 OF 01 #- #+
ANDJK BOOLEAN J K FLIP FLOP W AND INPUTS
BINARY01 BOOLEAN WESTINGHOUSE CIRCUIT (NOR GATES)
FLOP01 BOOLEAN FLIP FLOP (3NAND GATES)
3NAND BOOLEAN 3 INPUT NAND GATE
SELECT MAJOR OPTION WITH LIGHT PEN

Figure 4-18

In order to make a flip-flop out of his FLOP01 which is now a single 3NAND gate, he detects

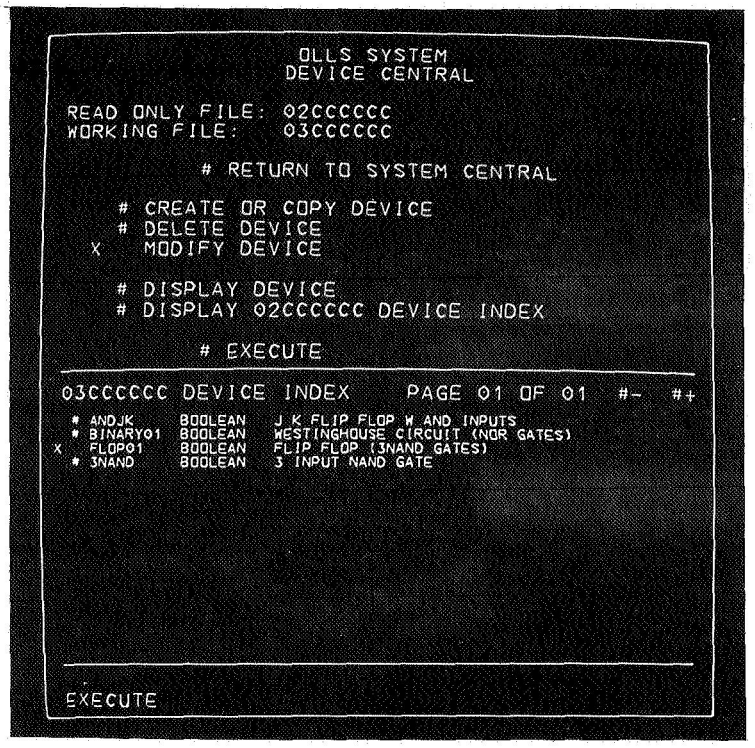
```
# MODIFY DEVICE
# FLOP01
```

followed by

```
# EXECUTE
```

Figure 4-19

The MODIFY DEVICE menu is initially very similar to the DISPLAY DEVICE menu with the addition of the four options which allow the designer to modify the device size, shape, terminals, or equations.


OLLS SYSTEM
DEVICE CENTRAL
READ ONLY FILE: 02CCCCCC
WORKING FILE: 03CCCCCC
RETURN TO SYSTEM CENTRAL
CREATE OR COPY DEVICE
DELETE DEVICE
X MODIFY DEVICE
DISPLAY DEVICE
DISPLAY 02CCCCCC DEVICE INDEX
EXECUTE
03CCCCCC DEVICE INDEX PAGE 01 OF 01 #- #+
ANDJK BOOLEAN J K FLIP FLOP W AND INPUTS
BINARY01 BOOLEAN WESTINGHOUSE CIRCUIT (NOR GATES)
X FLOP01 BOOLEAN FLIP FLOP (3NAND GATES)
3NAND BOOLEAN 3 INPUT NAND GATE
EXECUTE

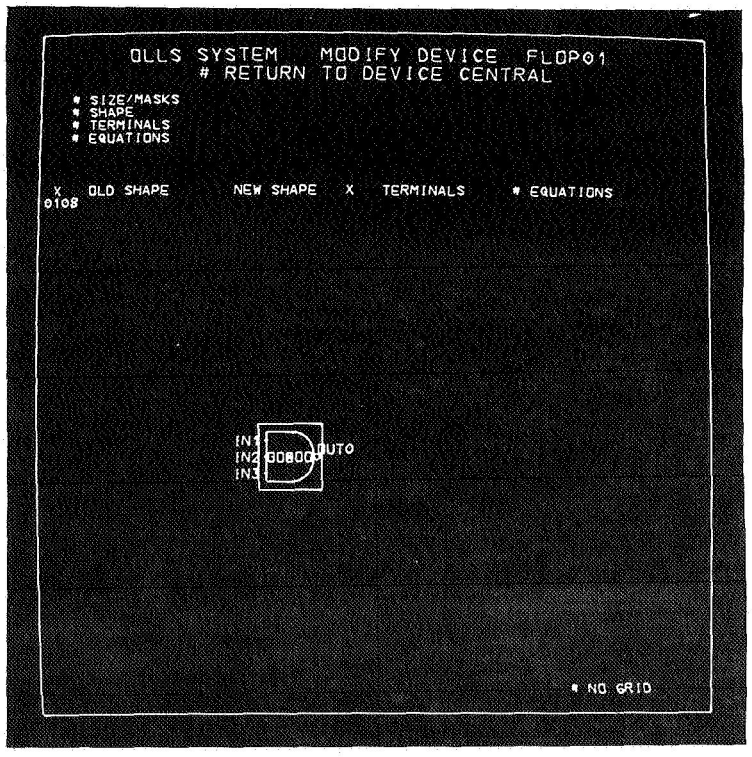

OLLS SYSTEM MODIFY DEVICE FLOP01
RETURN TO DEVICE CENTRAL
SIZE/MASKS
SHAPE
TERMINALS
EQUATIONS
X OLD SHAPE NEW SHAPE X TERMINALS # EQUATIONS
0108
IN1
IN2
IN3
D
AUTO
NO GRID

Figure 4-20

A flip-flop is larger than a 3NAND gate so the designer first detects

SIZE/MASKS

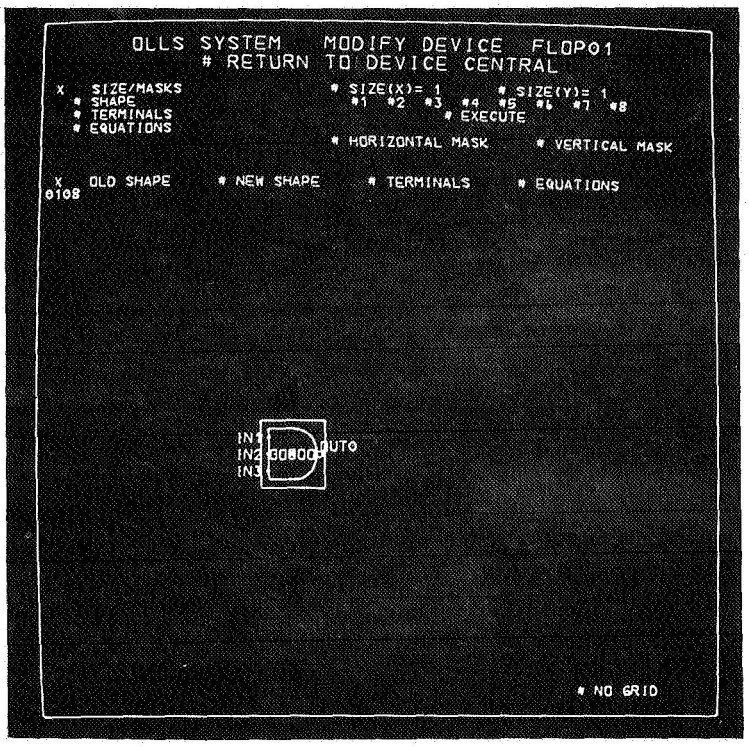
and a sub menu appears which allows the designer to alter the size. (MASKS will be explained later.)

Figure 4-21

The designer detects

SIZE(Y) = 1
3
EXECUTE

to change the vertical size to 3 units. (The #NO GRID - #GRID option at the bottom of the screen controls the display of the size outline.)


OLLS SYSTEM MODIFY DEVICE FLOP01
RETURN TO DEVICE CENTRAL
X SIZE/MASKS
SHAPE
TERMINALS
EQUATIONS
SIZE(X)= 1
#1 #2 #3 #4 #5 #6 #7 #8
SIZE(Y)= 1
EXECUTE
HORIZONTAL MASK
VERTICAL MASK
X OLD SHAPE
0108
NEW SHAPE
TERMINALS
EQUATIONS
IN1
IN2
IN3
OUTO
NO GRID

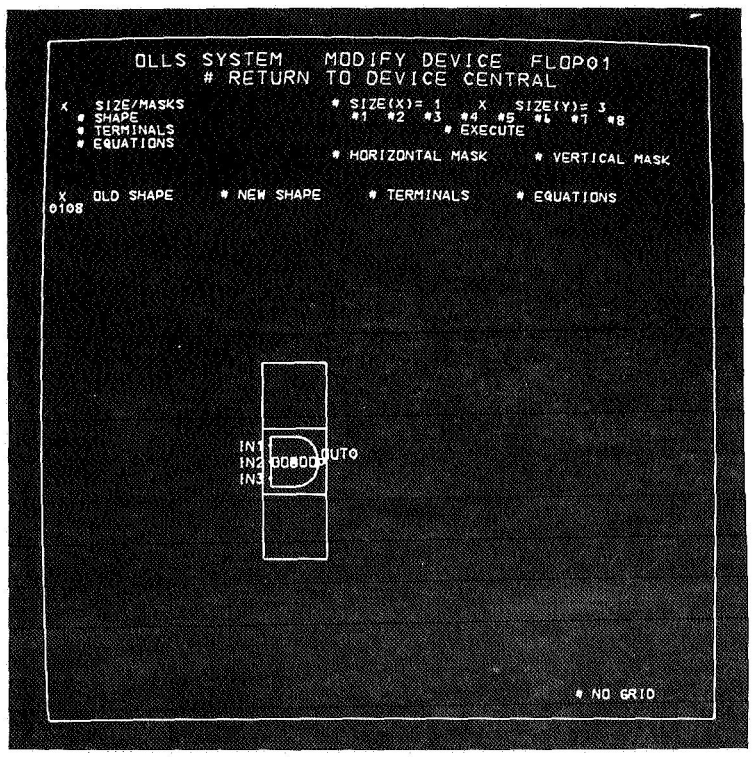

OLLS SYSTEM MODIFY DEVICE FLOP01
RETURN TO DEVICE CENTRAL
X SIZE/MASKS
SHAPE
TERMINALS
EQUATIONS
SIZE(X)= 1
#1 #2 #3 #4 #5 #6 #7 #8
X SIZE(Y)= 3
EXECUTE
HORIZONTAL MASK
VERTICAL MASK
X OLD SHAPE
0108
NEW SHAPE
TERMINALS
EQUATIONS
IN1
IN2
IN3
OUTO
NO GRID

Figure 4-22

The designer detects

SHAPE

to cause this sub menu to appear. The options shown affect the NEW SHAPE only and the OLD SHAPE (the permanent copy for the file) is not updated until # KEEP NEW SHAPE is detected.

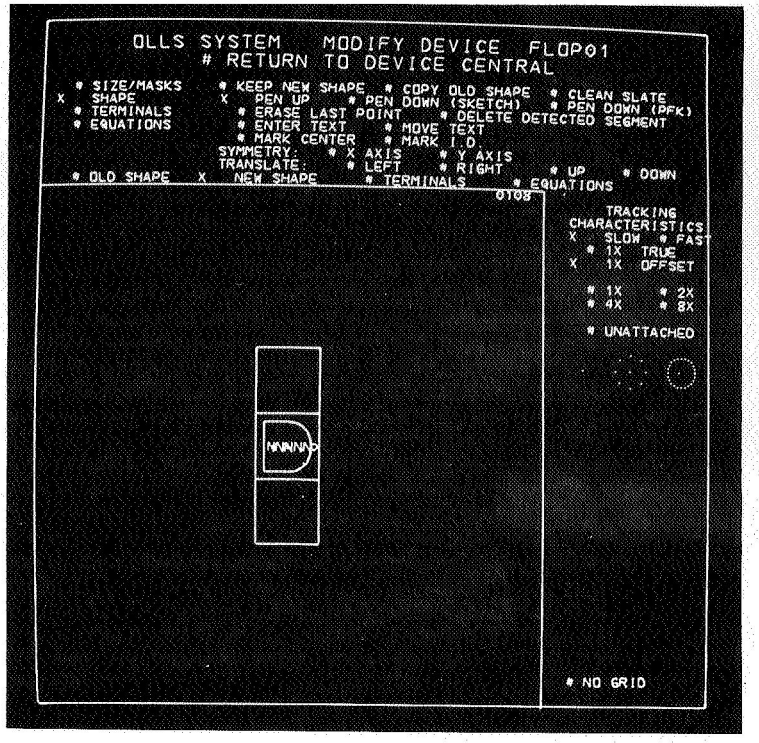
Figure 4-23

The designer rotates the center device to each of the two outer positions as shown by

a) # X AXIS, Symmetry
b) Move line of symmetry with tracking symbol,
c) detect segments of center device,
d) move line of symmetry again
e) detect segments of center device.

The photo shows the line of symmetry in position for steps d) and e) above.

The same result could have been obtained by TRANSLATE.


OLLS SYSTEM MODIFY DEVICE FLOP01
RETURN TO DEVICE CENTRAL
* SIZE/MASKS
X SHAPE
* TERMINALS
* EQUATIONS
* KEEP NEW SHAPE
X PEN UP
* ERASE LAST POINT
* ENTER TEXT
* MARK CENTER
* COPY OLD SHAPE
* PEN DOWN (SKETCH)
* MOVE TEXT
* MARK I.D.
* DELETE DETECTED SEGMENT
SYMMETRY: * X AXIS * Y AXIS
TRANSLATE: * LEFT * RIGHT * UP * DOWN
* OLD SHAPE X NEW SHAPE * TERMINALS * EQUATIONS
0108
TRACKING CHARACTERISTICS
X * SLOW * FAST
* 1X TRUE
X 1X OFFSET
* 1X * 2X
* 4X * 8X
* UNATTACHED
* NO GRID

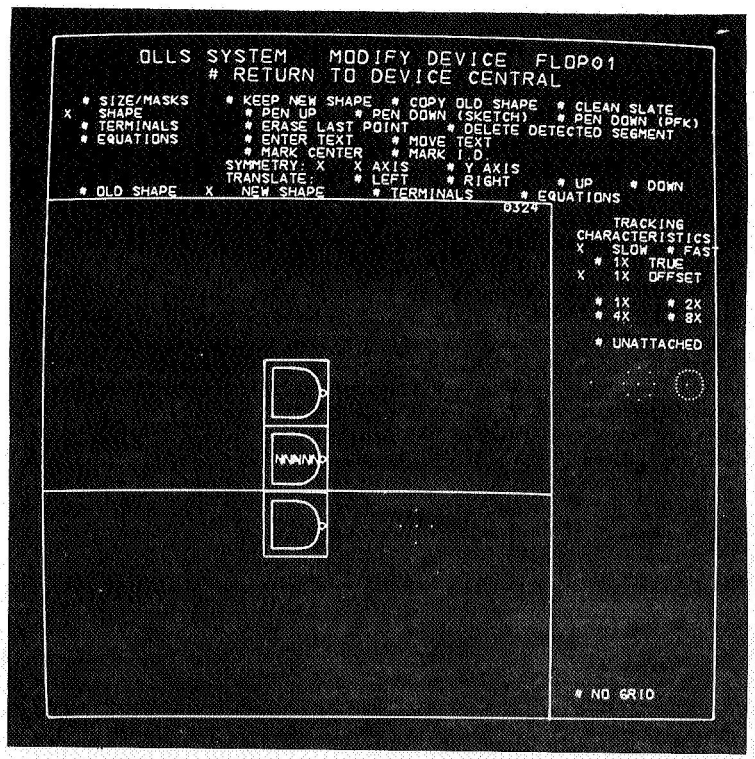

OLLS SYSTEM MODIFY DEVICE FLOP01
RETURN TO DEVICE CENTRAL
* SIZE/MASKS
X SHAPE
* TERMINALS
* EQUATIONS
* KEEP NEW SHAPE
* PEN UP
* ERASE LAST POINT
* ENTER TEXT
* MARK CENTER
* COPY OLD SHAPE
* PEN DOWN (SKETCH)
* MOVE TEXT
* MARK I.D.
* DELETE DETECTED SEGMENT
SYMMETRY: X X AXIS * Y AXIS
TRANSLATE: * LEFT * RIGHT * UP * DOWN
* OLD SHAPE X NEW SHAPE * TERMINALS * EQUATIONS
0324
TRACKING CHARACTERISTICS
X * SLOW * FAST
* 1X TRUE
X 1X OFFSET
* 1X * 2X
* 4X * 8X
* UNATTACHED
* NO GRID

Figure 4-24

The center device is deleted by

DELETE DETECTED SEGMENT

followed by detecting segments of the center device.

NOTE: The OLD SHAPE has not been affected by these operations. # OLD SHAPE will still display the original 3NAND and # COPY OLD SHAPE will restore the NEW SHAPE to the original 3NAND.

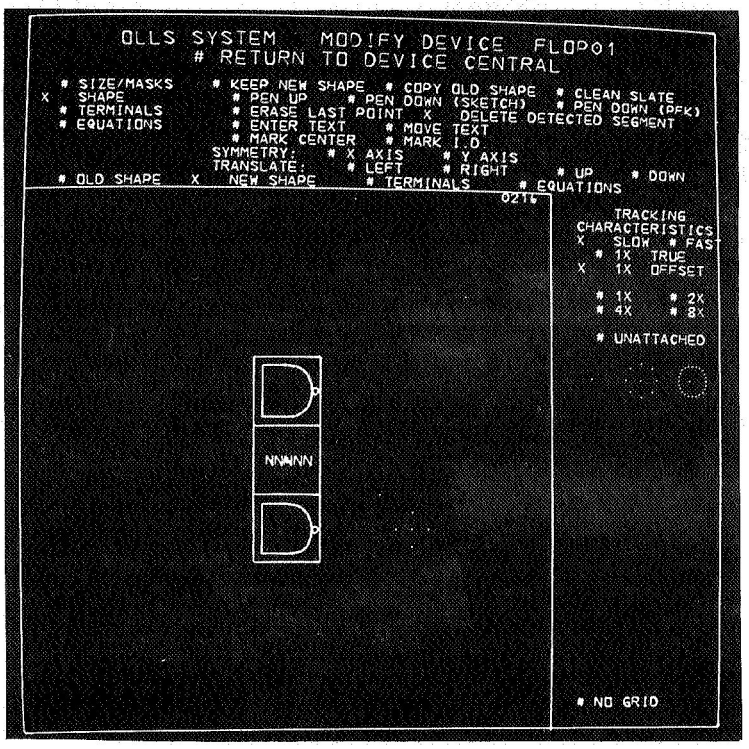
Figure 4-25

New segments are added to NEW SHAPE by

PEN DOWN (PFK)

moving the tracking symbol to each new position and marking each point with the programmed function keyboard. This continues until # PEN UP. This mode is useful for drawing long straight lines.

NOTE: # PEN DOWN (SKETCH) operates the same except points are added for every tracking symbol detect. This is useful for drawing small curved segments.


OLLS SYSTEM MODIFY DEVICE FLOP01
RETURN TO DEVICE CENTRAL
SIZE/MASKS
X # SHAPE
TERMINALS
EQUATIONS
KEEP NEW SHAPE
PEN UP
ERASE LAST POINT
ENTER TEXT
MARK CENTER
COPY OLD SHAPE
PEN DOWN (SKETCH)
X # DELETE DETECTED SEGMENT
MOVE TEXT
MARK I.D
SYMMETRY: # X AXIS # Y AXIS
TRANSLATE: # LEFT # RIGHT # UP # DOWN
OLD SHAPE X # NEW SHAPE # TERMINALS # EQUATIONS
0214
TRACKING CHARACTERISTICS
X # SLOW # FAST
1X TRUE
X # 1X OFFSET
1X # 2X
4X # 8X
UNATTACHED
NO GRID

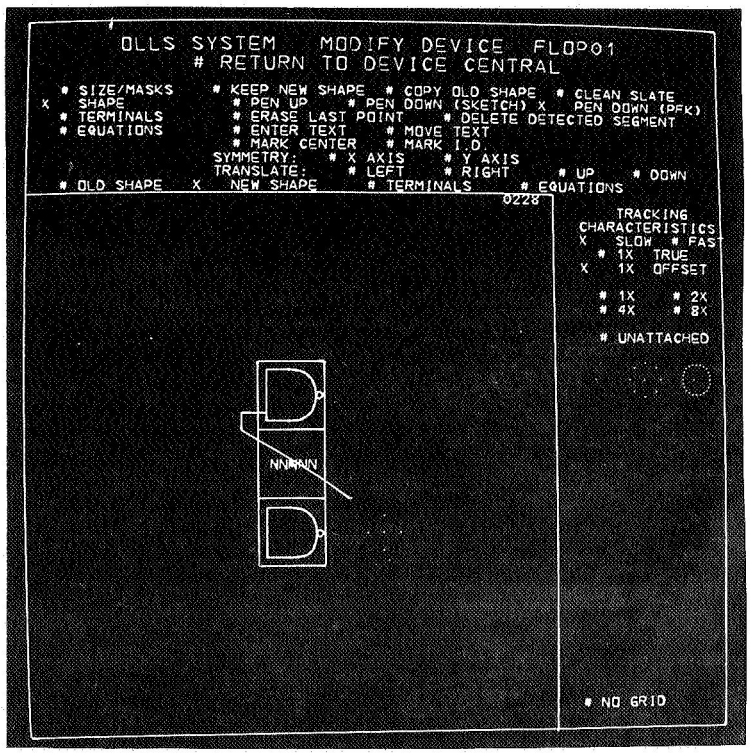

OLLS SYSTEM MODIFY DEVICE FLOP01
RETURN TO DEVICE CENTRAL
SIZE/MASKS
X # SHAPE
TERMINALS
EQUATIONS
KEEP NEW SHAPE
PEN UP
ERASE LAST POINT
ENTER TEXT
MARK CENTER
COPY OLD SHAPE
PEN DOWN (SKETCH)
X # DELETE DETECTED SEGMENT
MOVE TEXT
MARK I.D
SYMMETRY: # X AXIS # Y AXIS
TRANSLATE: # LEFT # RIGHT # UP # DOWN
OLD SHAPE X # NEW SHAPE # TERMINALS # EQUATIONS
0228
TRACKING CHARACTERISTICS
X # SLOW # FAST
1X TRUE
X # 1X OFFSET
1X # 2X
4X # 8X
UNATTACHED
NO GRID

Figure 4-26

The final segment is added again by

X AXIS Symmetry

and

KEEP NEW SHAPE

is detected.

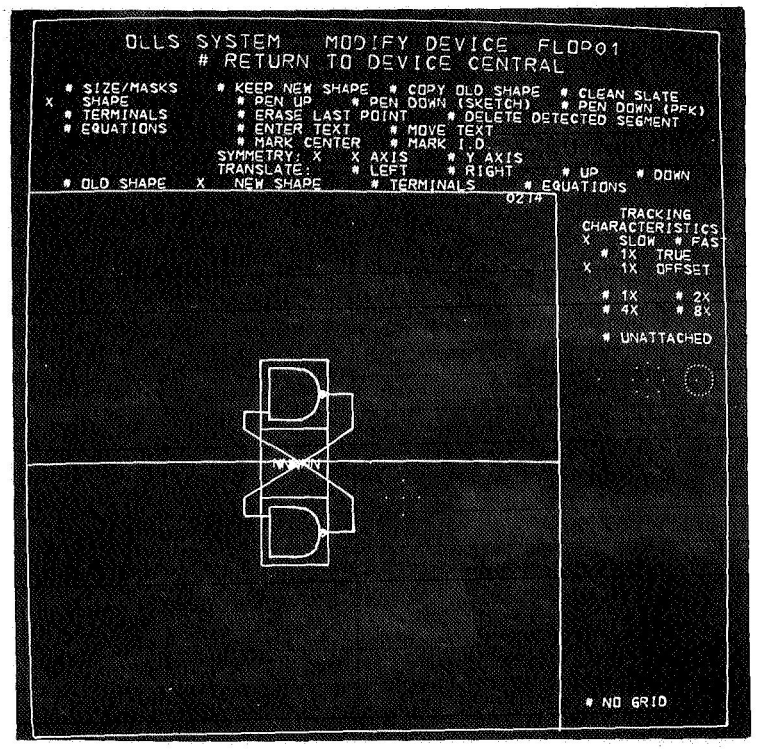
NOTE: Many useful features have not been shown (such as tracking magnification) since they do not lend themselves readily to still photography.

Figure 4-27

To modify the original 3NAND terminals, detect

TERMINALS

to cause this sub menu to appear. The terminals shown are of the original 3NAND gate. We will delete IN3, move OUT0, IN1, and IN2, and finally add a new terminal OUT4.


OLLS SYSTEM MODIFY DEVICE FLOP01
RETURN TO DEVICE CENTRAL
SIZE/MASKS
X SHAPE
TERMINALS
EQUATIONS
KEEP NEW SHAPE
PEN UP
ERASE LAST POINT
ENTER TEXT
MARK CENTER
COPY OLD SHAPE
PEN DOWN (SKETCH)
MOVE TEXT
MARK I.O.
CLEAN SLATE
PEN DOWN (PFK)
DELETE DETECTED SEGMENT
SYMMETRY: X X AXIS Y AXIS
TRANSLATE: # LEFT # RIGHT # UP # DOWN
OLD SHAPE X NEW SHAPE # TERMINALS # EQUATIONS
0214
TRACKING CHARACTERISTICS
X # SLOW # FAS
1X TRUE
X 1X OFFSET
1X # 2X
4X # 8X
UNATTACHED
NO GRID

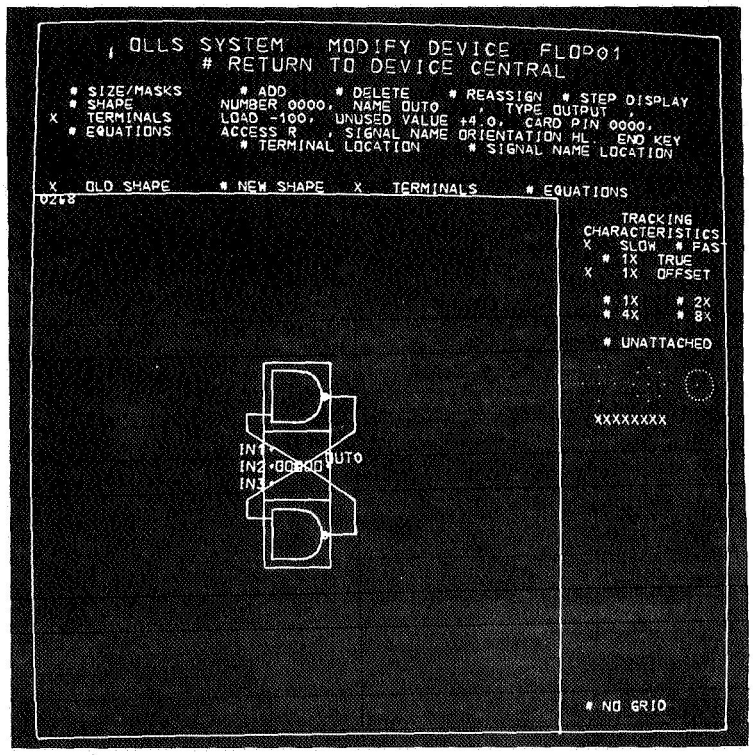

OLLS SYSTEM MODIFY DEVICE FLOP01
RETURN TO DEVICE CENTRAL
SIZE/MASKS
SHAPE
X TERMINALS
EQUATIONS
ADD # DELETE # REASSIGN # STEP DISPLAY
NUMBER 0000, NAME OUTO , TYPE OUTPUT ,
LOAD -100, UNUSED VALUE +4.0, CARD PIN 0000,
ACCESS R , SIGNAL NAME ORIENTATION HL END KEY
TERMINAL LOCATION # SIGNAL NAME LOCATION
X OLD SHAPE # NEW SHAPE X TERMINALS # EQUATIONS
0268
TRACKING CHARACTERISTICS
X # SLOW # FAS
1X TRUE
X 1X OFFSET
1X # 2X
4X # 8X
UNATTACHED
XXXXXXXX
IN1
IN2 .00000 UTO
IN3
NO GRID

Figure 4-28

Terminal IN3 was deleted by detecting

DELETE,

detecting terminal IN3 on the device, and pressing the END KEY.

The other three terminals were repositioned using

TERMINAL LOCATION
and
SIGNAL NAME LOCATION

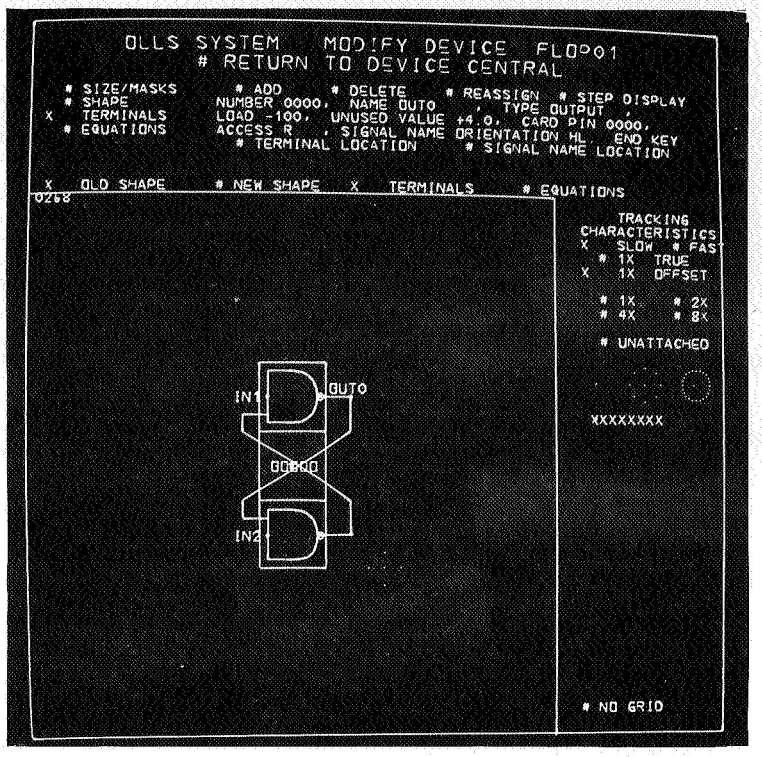
Figure 4-29

A new terminal OUT4 is added by detecting

ADD,

typing in the terminal characteristics as desired, and pressing the END KEY.

Some checks are automatically made in the background such as for non-unique terminal numbers or names etc.


OLLS SYSTEM MODIFY DEVICE FLOP01
RETURN TO DEVICE CENTRAL
SIZE/MASKS
SHAPE
X TERMINALS
EQUATIONS
ADD # DELETE # REASSIGN # STEP DISPLAY
NUMBER 0000, NAME OUT0 , TYPE OUTPUT ,
LOAD -100, UNUSED VALUE +4.0, CARD PIN 0000,
ACCESS R , SIGNAL NAME ORIENTATION HL END KEY
TERMINAL LOCATION # SIGNAL NAME LOCATION
X OLD SHAPE # NEW SHAPE X TERMINALS # EQUATIONS
0268
TRACKING CHARACTERISTICS
X SLOW # FAST
1X TRUE
X 1X OFFSET
1X # 2X
4X # 8X
UNATTACHED
XXXXXXXX
IN1
OUT0
IN2
NO GRID

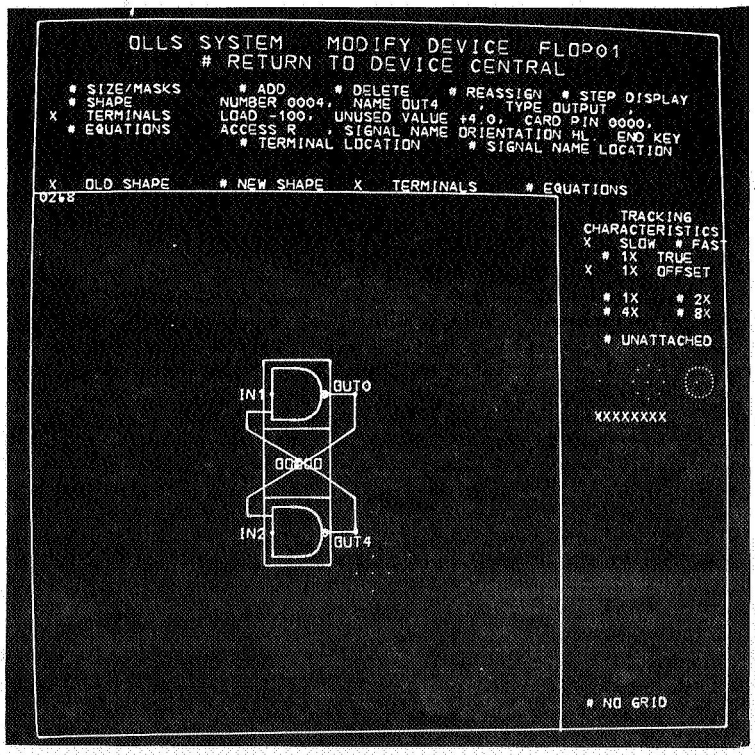

OLLS SYSTEM MODIFY DEVICE FLOP01
RETURN TO DEVICE CENTRAL
SIZE/MASKS
SHAPE
X TERMINALS
EQUATIONS
ADD # DELETE # REASSIGN # STEP DISPLAY
NUMBER 0004, NAME OUT4 , TYPE OUTPUT ,
LOAD -100, UNUSED VALUE +4.0, CARD PIN 0000,
ACCESS R , SIGNAL NAME ORIENTATION HL END KEY
TERMINAL LOCATION # SIGNAL NAME LOCATION
X OLD SHAPE # NEW SHAPE X TERMINALS # EQUATIONS
0268
TRACKING CHARACTERISTICS
X SLOW # FAST
1X TRUE
X 1X OFFSET
1X # 2X
4X # 8X
UNATTACHED
XXXXXXXX
IN1
OUT0
IN2
OUT4
NO GRID

Figure 4-30

To modify the original 3NAND equation, detect

EQUATIONS

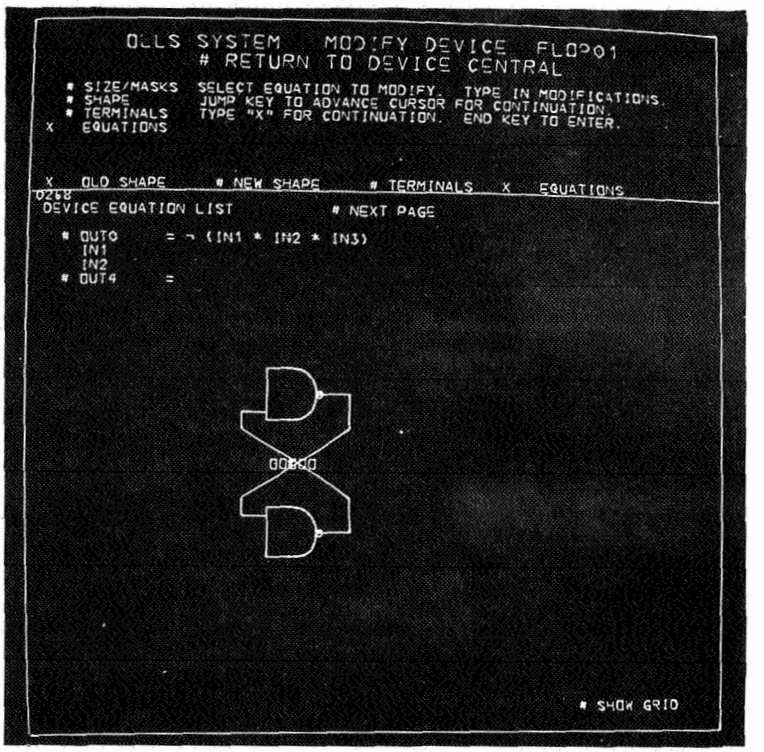
causes this sub menu to appear.

The equation for OUT0 is the original form from the 3NAND. Only terminals defined as "OUTPUT" or "INT. VAR." have a "#" and are alterable.

Figure 4-31

The new equations have been entered as per the instructions on the screen.
(Since no delays were specified, a delay of 1 unit is assumed.)

NOTE: It is at this point in time that the equations which describe a device can be compiled into machine code for the simulation routines. The manner in which the equations should be interpreted is decided by the "Device Class" field when the device was created. Equations for an analog device should perhaps be delivered to the FORTRAN Compiler while equations for a boolean device should be delivered to another compiler.


OLLS SYSTEM MODIFY DEVICE FLOP01
RETURN TO DEVICE CENTRAL
SIZE/MASKS SELECT EQUATION TO MODIFY. TYPE IN MODIFICATIONS.
SHAPE JUMP KEY TO ADVANCE CURSOR FOR CONTINUATION.
TERMINALS TYPE "X" FOR CONTINUATION. END KEY TO ENTER.
X EQUATIONS
X OLD SHAPE # NEW SHAPE # TERMINALS X EQUATIONS
0268
DEVICE EQUATION LIST # NEXT PAGE
OUT0 = ¬ (IN1 * IN2 * IN3)
IN1
IN2
OUT4 =
SHOW GRID

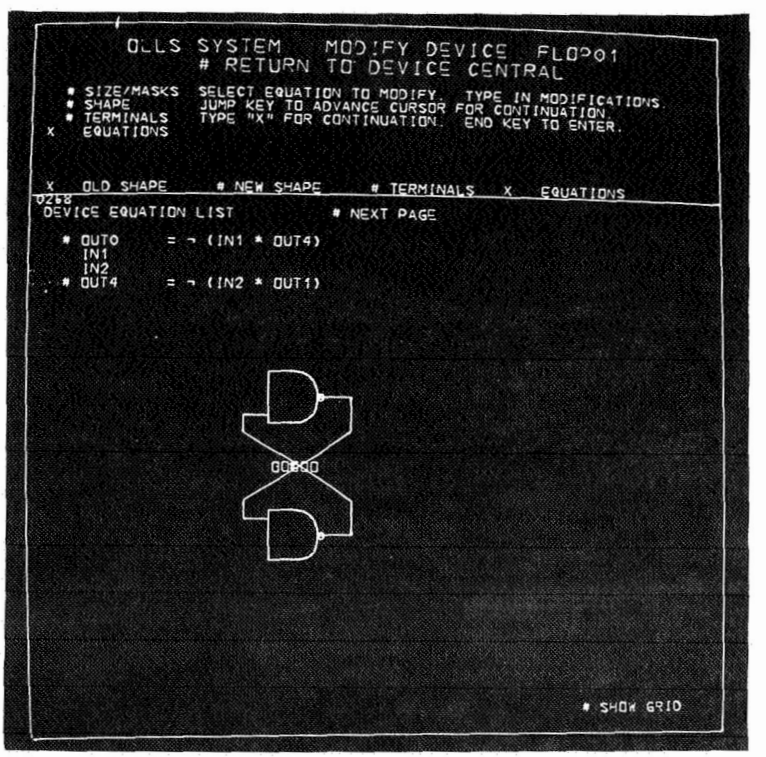

OLLS SYSTEM MODIFY DEVICE FLOP01
RETURN TO DEVICE CENTRAL
SIZE/MASKS SELECT EQUATION TO MODIFY. TYPE IN MODIFICATIONS.
SHAPE JUMP KEY TO ADVANCE CURSOR FOR CONTINUATION.
TERMINALS TYPE "X" FOR CONTINUATION. END KEY TO ENTER.
X EQUATIONS
X OLD SHAPE # NEW SHAPE # TERMINALS X EQUATIONS
0268
DEVICE EQUATION LIST # NEXT PAGE
OUT0 = ¬ (IN1 * OUT4)
IN1
IN2
OUT4 = ¬ (IN2 * OUT1)
SHOW GRID

Figure 4-32

Satisfied with the definition of FLOP01, our designer has detected

RETURN TO DEVICE CENTRAL

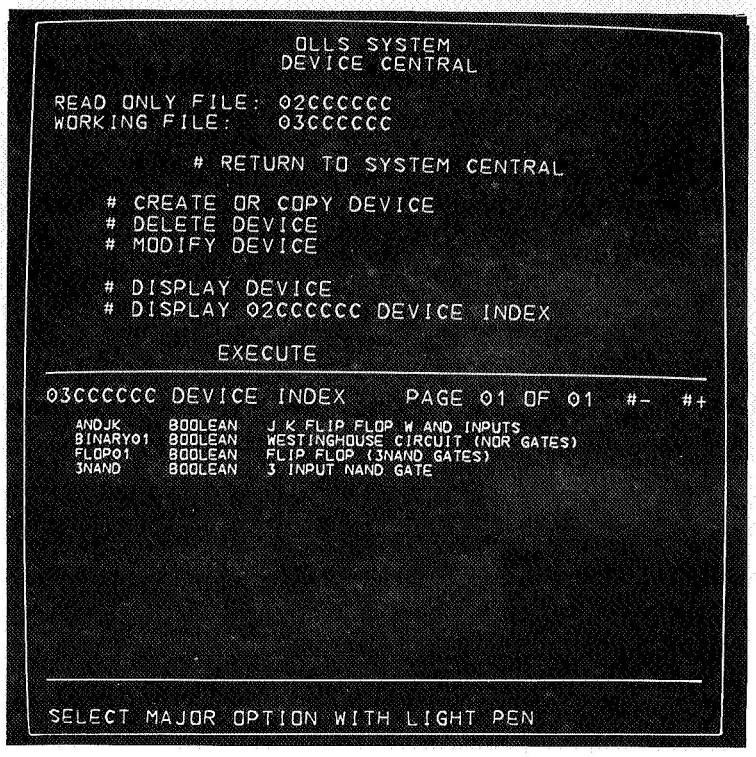
Satisfied with the list of available devices, our designer will detect

RETURN TO SYSTEM CENTRAL

Figure 4-33

Again we see the major options available. We have a satisfactory list of devices so we can proceed to:

DRAWING MANIPULATE


OLLS SYSTEM
DEVICE CENTRAL
READ ONLY FILE: 02CCCCCC
WORKING FILE: 03CCCCCC
RETURN TO SYSTEM CENTRAL
CREATE OR COPY DEVICE
DELETE DEVICE
MODIFY DEVICE
DISPLAY DEVICE
DISPLAY 02CCCCCC DEVICE INDEX
EXECUTE
03CCCCCC DEVICE INDEX PAGE 01 OF 01 #- #+
ANDJK BOOLEAN J K FLIP FLOP W AND INPUTS
BINARY01 BOOLEAN WESTINGHOUSE CIRCUIT (NOR GATES)
FLOP01 BOOLEAN FLIP FLOP (3NAND GATES)
3NAND BOOLEAN 3 INPUT NAND GATE
SELECT MAJOR OPTION WITH LIGHT PEN

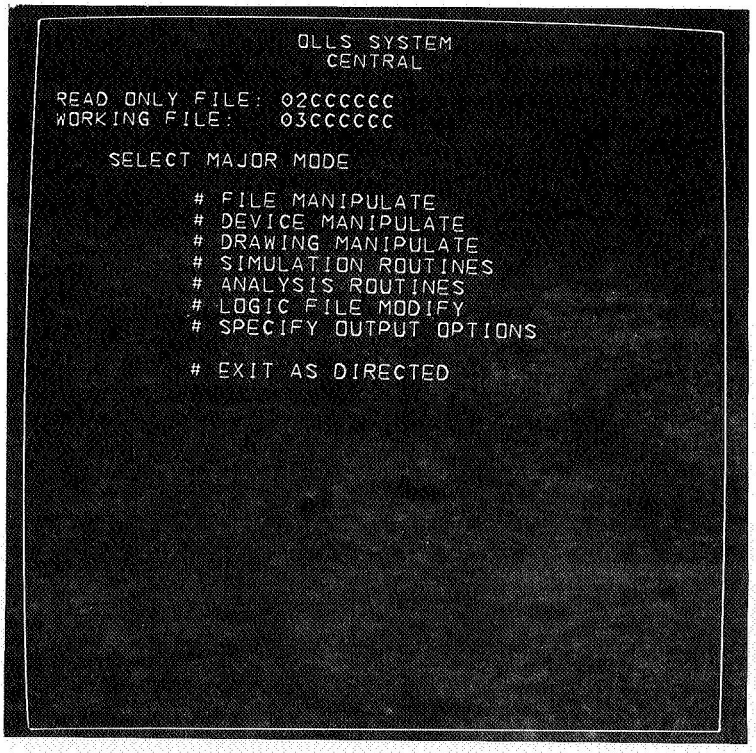

OLLS SYSTEM
CENTRAL
READ ONLY FILE: 02CCCCCC
WORKING FILE: 03CCCCCC
SELECT MAJOR MODE
FILE MANIPULATE
DEVICE MANIPULATE
DRAWING MANIPULATE
SIMULATION ROUTINES
ANALYSIS ROUTINES
LOGIC FILE MODIFY
SPECIFY OUTPUT OPTIONS
EXIT AS DIRECTED

Figure 4-34

As of this writing

DRAWING CENTRAL

has been fully implemented and checked out. It operates in much the same fashion as DEVICE CENTRAL shown earlier.

Figure 4-35

As of this writing

MODIFY DRAWING
DISPLAY DRAWING
and
OUTPUT OPTIONS

are in final stages of implementation.
(See Fig. 3-41 and Fig. 3-42 in E-2265.)

5. CONCLUSION

After almost three years of experimenting with different file structures (from the very earliest forms in Honeywell 1800 MAC to the very elegant forms possible in FILLIP), with different drawing and simulation algorithms, and with both interactive and non-interactive systems, we believe the following three points express the essence of our findings and our contributions to the area of computer aided design of logic circuits.

a) The nature of the OLLS project is of such complexity that the key to success in any implementation of the required goals is the structure of the data file. The means of the implementation of this structure will affect only the running efficiency and ease of coding. The data structure outlines in Section 3.1 of E-2265 embodies (we feel) most of the requirements of today's logic subsystems and has the ability to be easily expanded as future developments in this field require.

b) A card input system is almost unacceptable in a production environment. As an experimental tool, however, a card input system such as the FILLIP logical simulator can be a powerful engineering aid to understanding complex logic circuits.

c) Today's technology in the field of logic circuits is advancing too fast for a system which has not been designed to accommodate changes with ease. It is absolutely essential (especially in the area of defining devices) that there be a minimum of arbitrary constraints upon the user such as insufficient file size, limited number and variety of device terminals, shapes, or behavior, or input/output formats which are not completely flexible to suit the current application of the system.

We now have an operational facility (MAC 360) to produce logic flow diagrams, logical simulations, useful output lists, and a wrap deck compatible with existing wirewrap software. We have the complete FILLIP Card Input System almost operational as described in Section 3. We will continue our efforts at debugging some of the newer features of that system although at a reduced level since its production usefulness is not as apparent now as is the CRT system. Most of our effort in the near future will be toward making the interactive CRT system operational, at least to the point where we can produce logic flow diagrams, lists, and

plots which are of superior quality and less expensive in terms of time and money than the MAC 360 system. We anticipate being in a position to phase out the MAC 360 system in early 1969.

For purely economic reasons we have fallen short of our goal: We do not have nor will we have an On Line Logical Simulator as the acronym OLLS suggests. We have instead two separate systems which if combined would do all that we initially set out to do. We have the on-line CRT system which has full drawing, file manipulating, and output capability but which cannot at this time perform logical simulations; and we have the FILLIP card input system which is not interactive, cannot produce any graphic output, but probably has the most powerful logical simulation capability available in the country today. Most of the difficult design work for an integrated system is complete, however, and we are now in a position where we require some external support for the implementation of the total system. The choice of whether it should be an all FILLIP implementation, a machine language implementation, or a combination, is not important. What is important is that we have shown the utility of such a system, have explored most of the alternatives which might be considered in implementing it, and have outlined a clear set of input, output, and internal structure requirements necessary for the computer aided design of logic systems.

E-2410

DISTRIBUTION LIST

Internal

R. Battin
R. Crisp
E. Duggan
J. B. Feldman
S. Forter
F. Glick
A. Green
W. Grigg
Eldon Hall
A. Harano
D. Hoag
A. Hopkins (25)
F. Houston
H. R. Howie (25)
*J. Kingston
A. Kosmala
A. Laats
J. Laning
L. Larson
P. Mimno
J. Nevins
J. Nugent
R. Ragan
G. Schwartz
H. Thaler
M. Trageser
R. Woodbury
W. Wrigley
Apollo Library (2)
MIT/IL Library (6)

*Letter of transmittal only

External

NASA/ERC (50 + 1R)
575 Technology Square
Cambridge, Massachusetts

ATTN: KC/Computer Research Laboratory
Mr. D. J. Kelleher (Letter of Transmittal only)