1. IDENTIFICATION

1.1 Maindec 712
1.2 PDP-7 Tape Punch (Type 75D) Test
1.3 July 22, 1965
2. ABSTRACT

Maindec 712 incorporates three separate programs for testing the functions of the paper tape punch. All three programs are in the computer at the same time. The first program checks the 1-second delay for proper operation when the initial Punch Select IOT is given. The second program is designed for use with a CRT. It causes the contents of the AC Switches to be punched continuously on tape in either binary or alphanumeric modes. The third program causes the punch to produce a tape containing a sequence of pseudo-random numbers; this tape is then checked using the tape reader.

3. REQUIREMENTS

3.1 Storage

Maindec 712 occupies memory registers 00000-01423 including output subroutines.

3.2 Subprograms

Maindec 712 uses the Teletype Output Package, Digital-7-10-0, and the Octal Print Subroutine, Digital-7-14-0. Both of these subroutines are included on the program tape.

3.3 Equipment

Standard PDP-7 with Paper Tape Reader and Punch

3.4 Miscellaneous

The following tapes are supplied:

ASCII (2, labeled A and D)

FF

ASCII Tapes of the two library subroutines (see Section 3.2) are not provided.

4. USAGE

4.1 Loading

The RIM Loader must be in memory.

4.1.1 Set the ADDRESS switches to 17770.

4.1.2 Place the FF Program tape in the reader.

4.1.3 Press START.

4.3 Switch Settings

4.3.1 Loading Address

17770

4.3.2 Starting Addresses

00040 (Part 1)
00100 (Part 2)
00200 (Part 3)
4.3.3 Other settings

The table below gives the various AC switch settings for Parts 2 and 3 of the Punch Test. Part 1 does not use the ACS.

<table>
<thead>
<tr>
<th>Test</th>
<th>Switches</th>
<th>Setting</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part 2</td>
<td>ACS₀</td>
<td>down</td>
<td>Punch alphanumeric.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>up</td>
<td>Punch binary.</td>
</tr>
<tr>
<td>Part 2</td>
<td>ACS₁</td>
<td>down</td>
<td>Punch character true.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>up</td>
<td>Alternate true character and complement.</td>
</tr>
<tr>
<td>Part 2</td>
<td>ACS₁₀⁻¹₇</td>
<td>---</td>
<td>Setting determines character to be punched.</td>
</tr>
<tr>
<td>Part 3</td>
<td>ACS₀</td>
<td>down</td>
<td>Halt on error.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>up</td>
<td>Do not halt on error.</td>
</tr>
<tr>
<td>Part 3</td>
<td>ACS₁</td>
<td>down</td>
<td>Print error messages.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>up</td>
<td>Do not print error messages.</td>
</tr>
<tr>
<td>Part 3</td>
<td>ACS₁₇</td>
<td>down</td>
<td>Punch random sequence.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>up</td>
<td>Terminate random punch and prepare for reader check.</td>
</tr>
</tbody>
</table>

4.4 Start up and Operation

4.4.1 Part 1 -- Startup Delay Test

Set the ADDRESS switches to 00040 and press START. The program title will be printed on the Teleprinter:

STARTUP DELAY TEST

Turn on punch power at console.

If the test is successful the message

DELAY O.K. TURN OFF PUNCH POWER SWITCH

is printed, and the computer halts with all 1's displayed in the AC lights.

4.4.2 Part 2 -- Scope Waveform Test

Set the ADDRESS switches to 00100 and press START. The program title and instructions are printed:

SCOPE WAVEFORM TEST.
ACS₀ UP FOR BINARY, DOWN FOR ALPHA.
ACS₁ UP IF COMPLEMENTING, DOWN IF NOT.
ACS₁₀⁻¹₇ FOR CHARACTER PUNCHED.

and the computer halts with the AC clear.
Set the ACS as desired, and press CONTINUE. The ACS setting can be changed at any time during the execution of the test; the program will respond immediately to the new setting. There is no final halt in this part; the operator must stop the computer manually.

4.4.3 Part 3 -- Random Sequence Punch

Set the ACS to 00200 and press START. The program title and instructions for the punch section of the test are printed:

```
RANDOM SEQUENCE PUNCH ROUTINE
ACS0 DOWN FOR ERROR HALTS, UP FOR NONE.
ACS1 DOWN FOR ERROR PRINT, UP FOR NONE.
ACS17 DOWN TO START. PUT UP TO STOP PUNCHING.
```

The computer halts with the AC and link clear.

Set the ACS in accordance with the instructions, and press CONTINUE. When as much tape as is desired has been produced, raise AC switch 17. The random sequence will stop, a short length of blank trailer is punched, and the title and instructions for the reader check are printed:

```
READER CHECK OF RANDOM TAPE
ACS0 AND ACS1 AS FOR PUNCH. ERRORS PRINTED BELOW.
CHARACTER
PUNCHED EXPECTED
```

The computer halts with the AC and link clear. Remove the random sequence tape from the punch bin, and place it in the reader so that the blank leader falls under the read heads. Set the ACS as desired, and press CONTINUE. Errors will be printed in columns under the headings PUNCHED and EXPECTED. When the whole tape has been scanned, the program comes to a final halt with the AC lights displaying all 1's.

4.5 Errors

Errors E00, E20, and E21 will always halt, regardless of the setting of ACS0. All other errors will stop only if ACS0 is down.

4.5.1 Error stops in Part 1

Error: E00
Message: STARTUP DELAY FAILURE. PRESS CONTINUE TO TRY AGAIN.
C(MA): 00064
C(AC): Time left in clock
Cause: The punch flag was set before about 1.1 seconds elapsed. This indicates a failure of the one-shot delay which prevents the first select instruction from taking effect until the drive clutch has been engaged. The AC contains the complement of the contents of register 00007. This number is the octal equivalent of the time remaining when the flag was set, in 60ths of a second.
Recovery: Press CONTINUE. The test will start over again.

4.5.2 Error stops in Part 2.

There are no error stops in this part of the test.

4.5.3 Error stops in Part 3.

Error: E01
Message: PSA+10 DID NOT CLEAR AC.
C(MA): 00153
C(AC): I/O status
Cause: The IOT instruction PSA+10 failed to clear the AC before selecting the punch. This test is made only when punching the blank leader and trailer on the test tape.
Recovery: Press CONTINUE. The test will proceed from the point of the error.

Error: E10
Message: CAF DID NOT CLEAR PUNCH FLAG.
C(MA): 00216
C(AC): I/O status
Cause: The punch flag was unaffected by the CAF instruction. This error implies that the punch flag is permanently on, since at any time prior to this point in the program, pressing START should have cleared the flag.
Recovery: Press CONTINUE. The test will proceed from the point of the error.

Error: E11
Message: PSF SKIPPED ON CLEARED FLAG.
C(MA): 00340
C(AC): I/O status
Cause: The PSF instruction skipped, even though the punch flag was clear.
Recovery: Press CONTINUE. The test will proceed from the point of the error.

Error: E12
Message: PSA DID NOT CLEAR PUNCH FLAG
C(MA): 00227
C(AC): I/O status
Cause: The punch flag was unaffected by the IOT. If it occurs before the test sequence has been scanned, this error, like E10, implies that the punch flag is somehow disconnected and permanently on. If it occurs during the reading of the test tape, however, it probably indicates a random failure.
Recovery: Press CONTINUE. The test will proceed from the point of the error.

Error: E13
Message: PSA FAILED TO SKIP WHEN FLAG WAS SET.
C(MA): 00030
C(AC): I/O status
Cause: This is the other half of the PSF test. In this case, it failed to skip at the proper time.
Recovery: Press CONTINUE. The test will proceed from the point of the error.

Error: E20
Message: NO INTERRUPT AFTER 16.1 MSEC. IS THE PUNCH REALLY ON?
C(MA): 00240
C(AC): I/O status
Cause: The time ran out before an interrupt occurred. The question in the diagnostic message implies that the proper program did not punch anything beyond the blank leader. If the error occurs after some of the sequence has been punched, the failure is probably in the interrupt control.
Recovery: Press CONTINUE. The test will begin again; the title and instructions are printed. When the computer halts, proceed as described in Section 4.4.3.
Error: E21
Message: INTERRUPT FROM SOMETHING OTHER THAN THE PUNCH. CHECK THE OTHER
DEVICES....AC SHOWS I/O STATUS. PRESS CONTINUE TO START ALL OVER AGAIN.
C(MA): 00035
C(AC): I/O status
Cause: An interrupt came from some other device than the punch. By examining the I/O status
in the AC lights, the offending device can be identified.
Recovery: Press CONTINUE. The test will begin again; the title and instructions are printed.
When the computer halts, proceed as described in Section 4.4.3.

Error: E30
Message: None
C(MA): 00325
C(AC): Bits in error
Cause: The character read from tape did not match the one calculated in the program. The bits
which did not match are shown in the AC lights as 1's. No message is printed, but the
number from the tape and the calculated number are printed in the columns labeled
"PUNCHED" and "EXPECTED", respectively.
Recovery: Press CONTINUE. The test will proceed from the point of the error.

4.6 Error Recovery
See Section 4.5, Errors.

5. RESTRICTIONS
Part 1 of this test will not work on any machine equipped with the automatic priority interrupt
option because the clock is not connected to register 00007 when the API is installed. The other two tests
will work as long as the API is kept off. Since pressing START always disables the API, there should be no
interference from it.

6. DESCRIPTION
6.1 Discussion
The three parts of Maindec 712 are all on the same program tape and exist in core together.
The operator may switch from one to the other at will, since none interferes with any other.
6.1.1 Part 1 -- Startup Delay Test
If the punch is selected after it has been lying idle, a delay prevents it from being enabled
until about 1 second has passed, in order to allow the drive mechanism to engage the clutch. The test
of part 1 times this delay; if the punch flag is set before about 1.1 seconds have elapsed, an error message
is printed, and the computer halts.

The timing is performed by presetting register 00007 and starting the clock. The punch is selected
by a PSA+10 instruction. A short sequence then tests the clock flag and the punch flag in order. As long
as the clock has not run out, the link remains off; if the punch flag is set in this interval, the error is
noted. When the clock runs out, the link is set; then when the punch flag is raised no error occurs. If
the test is successful, a message is printed and the computer stops.

6.1.2 Part 2 -- Scope Waveform Test
This program will punch the character represented by the setting of AC switches 10-17. Switches set up represent 1's. If ACS0 is up, the program will punch in binary mode the character represented by ACS12-17. If ACS1 is up, the program will punch first the character, then its complement, in the mode indicated by ACS0.

6.1.3 Part 3 -- Random Sequence Punch

In this part, a sequence of pseudo-random numbers is calculated and punched. The punching can be terminated at any time by raising ACS17. The program keeps a count of the numbers punched, and uses this to determine when the end of the tape is reached while scanning it for errors. The error checking routine uses the same subprogram to calculate each digit, and compares this with the number read from tape.

6.2 APPLICATIONS

6.2.1 Part 1

Part 1 is used when there is reason to suspect trouble in the one-shot startup delay or in the integrating one-shot which holds the punch motor on for 5 seconds after the last select instruction has been processed.

6.2.2 Part 2

Part 2 has several uses. In binary mode, it tests the ability of the punch control to force the proper state of channels 7 and 8. A visual check of the tape as it comes from the punch will show if the logic is functioning correctly.

The operator can visually check the spacing of characters on tape by setting the ACS to 377 and punching in alphanumeric mode. Variations in character spacing can be easily detected. Since every channel is punched, overlapping characters are immediately apparent. With an oscilloscope attached, the operator can examine waveforms generated, as a single character is punched over and over again.

Another symptom of punch malfunctioning appears when residual noise from the punching of one character causes a channel in the succeeding character to be punched, even though it is not conditioned by a 1 in the AC. By repeatedly punching a code and its complement in succession, this type of error can be detected quite readily. Normally, the state of each channel changes every time the punch is selected; the error is revealed when the same channel in adjacent characters is punched.

6.2.3 Part 3

If the action of a punch magnet is being influenced by the action of the ones adjacent to it, a spurious bit may occasionally be punched. The random sequence punched in the test eventually produces every possible combination of bits in a character. It also provides another test of residual noise (see Section 6.2.2) by producing all possible combinations of successive characters, something which is not provided, for example, by a strict numeric sequence. The random sequence is as long as the operator wishes to make it.

7. METHODS

The random sequence punch program keeps a count of the number of characters punched. This count is used by the test program to determine when the end of the sequence has been reached. This count is remembered, so that the same tape can be tested several times as long as the test program remains in memory. A new tape should be punched each time the program is loaded.

8. FORMAT (Not applicable.)
9. EXECUTION TIME
   Parts 2 and 3 are indefinite in length. Part 1 takes just over 1 second if the test is successful.

10. PROGRAMS

10.4 Program Listing

Maindec 712: PDP-7 PUNCH TEST

/IN THREE PARTS:
  1. STARTUP DELAY TEST
  2. SCOPE WAVEFORM TEST
  3. RANDOM SEQUENCE PUNCH

/PART 1. STARTUP DELAY TEST
40/

STADT,

TIN
LAW STIT /PRINT TITLE
TSR
LAM -55 /DELAY TIME
DAC 7
IUF
CLON
PSA+10
CLSF /IS TIME UP?
SKPVCLL /NO.
STL /YES, SET SIGNAL
PSF /PUNCH DONE?
JMP -.4 /NO.
SZL /YES, DID TIME RUN OUT?
JMP SOUT /YES, TEST OK.
CLOF /NO, PRINT ERROR MESSAGE
TIN
LAW EM00
TSR
LAC 7
E00,

HLTVCMA /ERROR, AC SHOWS TIME LEFT.
JMP STADT /TRY AGAIN

SOUT,

TIN
LAW SOK
TSR
A,

HLTVCCLC /OK HALT.
JMP -.1 /CAN'T GET OUT OF IT

/PART 2: SCOPE WAVEFORM TEST. PUNCHES C(AC).

100/

PACS,

TIN
LAW ATIT /PRINT TITLE AND INSTRUCTIONS
TSR
PART 3. RANDOM SEQUENCE PUNCH

200/

RAPT,    | DZM NONO               /INITIALIZE COUNTER,   |
         | LAC (1233)             /RN SEEDS,          |
         | DAC RAN1               |
         | LAC (7622)             |
         | DAC RAN2               |
         | LAC (FLEX CAF          /AND EM10 DIAGNOSTIC |
         | DAC EPT                |
         | TIN                    |
         | LAW PTIT               |
         | TSR                    |

212 / C, | HLT VCLA VCLL          /WAIT FOR ACS.        |
         | CLOF                   |
         | CAF                    /TRY CLEARING FLAGS |
         | JMS FLAGT              |

E10,     | HLT VSTL               /ERROR: CAF FAILED. AC SHOWS STATUS |
         | LAM = -300             |
         | JMS PFEE               /PUNCH LEADER    |
         | LAC (FLEX PSA          |
         | DAC EPT                /SET DIAGNOSTIC |

PRAN,    | JMS RANDY              /GET A NUMBER       |
         | ISZ NONO               |
         | PSA                    |
         | JMS FLAGT              |

E12,     | HLT VSTL               /ERROR: PSA FAILED. AC SHOWS STATUS |
         | ION                    |
         | LAM = -6000            /INTERRUPT TIMER: 16.1 MSEC. |
         | DAC APTEM              |
         | ISZ APTEM              |
         | JMP . -1               /WAIT LOOP           |

ER20,    | JMS RACS               /ERROR ROUTINE: NO INTERRUPT |

B,       | CLOF                   /WAIT FOR ACS SETTING |
         | HLT VCLA               |
         | LAS VCLL               |
         | SPA                    /WHICH MODE?      |
         | STL                    /BINARY         |

PALP,    | PSA                    /START PUNCH IN ALPHA |
         | SZL                    /WHICH MODE?      |
         | PSB                    /BINARY. OVERRIDE PSA. |
         | PSF                    |
         | JMP . -1               |
         | RAL                    /FORMAT SWITCH |
         | SMAVRAR                /COMPLEMENT?     |
         | JMP PACH               /
         | CMA                    |
         | JMP PALP               /

PAL2,    | /

/WHICH MODE?   |
/BINARY        |
/START PUNCH IN ALPHA |
/BINARY. OVERRIDE PSA. |
/FORMAT SWITCH |
/COMPLEMENT?   |
LAW EM20
NOP  /ALWAYS HALTS
E20,
HLTVSTL  /ERROR STOP. AC SHOWS STATUS
JMP RAPT
PNEX,
LAS  /COME HERE FROM INTERRUPT SERVICE ROUTINE
RAR
SNL  /STOP PUNCHING?
JMP PRAN  /NO
LAM -150  /YES: PUNCH TRAILER
IOF
JMS PFEE
PREC,
IN  /READER CHECK
LAW PTIT
TSR
D,
HLTVCLAVCLL  /WAIT FOR SWITCHES
LAC (1233  /INITIALIZE RNS.
DAC RAN1
LAC (7622
DAC RAN2
LAC NONO  /SET COUNTER
CMA
DAC NON2
ISZ NON2
RSA  /READ LEADER
RSF
JMP -.1
RRB
SZA  /LEADER DONE?
JMP RECH+4  /YES.
JMP -.6  /NO
RECH,
RSA  /READ A CHARACTER
RSF
JMP -.1
RRB
RECH+4,
DAC APTEM
JMS RANDY
SAD APTEM  /COMPARE FACT WITH THEORY
JMP COUNT  /FACT IS OK.
ER30,
JMS RACS  /FACT IS WRONG
JMP E30P  /GO PRINT NUMBERS
JMP E30H  /GO HALT
COUNT,
ISZ NON2  /COUNT NUMBERS
JMP RECH  /NOT DONE YET
F,
HLTVCLVCCLL  /FINAL HALT
JMP -.1  /ABSOLUTELY.
E30P,
LAC APTEM
JMS OPT  /PRINT FACT
TYT
LAC RAN1
AND (377
JMS OPT
JMP RAC2
E30H,
LAC RAN1
AND (377
XOR APTEM
E30,
HLTVSTL
JMP COUNT

/ THE ERROR DETECTIVE 

FLAGT,
0
IORS
/ EXAMINE STATUS
RTL
SPA
/ IS PUNCH FLAG CLEAR?
JMP ER10
/ NO.
PSF
/ YES. TEST SKIP
JMP FLOUT
/ OK.
ER11,
JMS RACS
/ SKIP FAILED
LAW EM11
E11,
HLTVSTL
JMP FLOUT
ER10,
JMS RACS
/ FLAG NOT CLEAR
JMP F10P
/ GO PRINT MESSAGE
JMP I FLAGT
/ HERE IF HALT
FLOUT,
ISZ FLAGT
/ HERE IF NONE
JMP I FLAGT
E10P,
LAC EPT
/ CAF OR PSA
TY3
LAW EM10
JMP RAC2-1

/ A NECESSARY ITEM 

OTY,
0
/ CHAR PRINT SUBROUTINE
TLS
TSF
JMP .-1
JMP I OTY

/ SOME PUNCHY SUBROUTINES 

PAL2+1/

RANDY,
0
/ RANDOM NUMBER GENERATOR
CLL
LAC RAN1
RTL
ADD RAN2
DAC RAN1
RTL
ADD RAN2
RTL
DAC RAN2
LAC RAN1
AND (377
JMP I RANDY

PFEE, 0
DAC APTEM

PFEL, PSA+10
SZA
JMP ER01
PSF
JMP .-1

PFEL+5, ISZ APTEM
JMP PFFL
JMP I PFEE

ER01, JMS RACS
LAW EM01
E01, HLTvSTL
JMP PFEL+5

RACS, 0
IOF
LAS
RAL
SPAVRAR
JMP RAC2+1
TIN
XCT I RACS
TSR

RAC2,
LAS
SPA
ISZ RACS
ISZ RACS
IORS
JMP I RACS

THE INTERRUPT SERVICE ROUTINE

0/ 0
IORS
RTL
SMA
JMP ER21
JMP ISR

20/ 0
HLTvCLC
JMP .-1

/THIS IS IT

/TAPE FEEDER
/FEED LOOP
/TEST: IS AC CLEAR?
/NO
/YES.

/ERROR ROUTINE
/ERROR: AC NOT CLEARED

/THE ACS EXAMINER
/PRINT MESSAGE?
/NO

/WILL BE EITHER LAW OR JMP

/STOP FOR ERROR?
/NO

/IS PUNCH FLAG SET?
/NO, SPURIOUS BREAK.
/YES.

/CALCATCHER
ISR, PSF /TEST SKIP
     SKP /FAILED.
     JMP PNEX /OK.
ER13, JMS RACS /PSF FAILED.
     LAW EM13
E13, HLT VSTL /ERROR: AC SHOWS STATUS
     JMP PNEX
ER21, JMS RACS /SPURIOUS INTERRUPT
     LAW EM21
E21, NOP /ALWAYS HALTS
     HLT VSTL /AC SHOWS STATUS
     JMP RAPT
NON2, 0 /COUNTER FOR PART 3

/ERROR MESSAGES, NOTICES, ORACLES AND OTHER WISDOM
/TITLES AND INSTRUCTIONS
STII, TEXT -STARTUP DELAY TEST
      TURN ON PUNCH POWER AT CONSOLE.
-      
ATIT, TEXT /SCOPE WAVEFORM TEST.
     ACS0 UP FOR BINARY, DOWN FOR ALPHA.
     ACS1 UP IF COMPLEMENTING, DOWN IF NOT.
     ACS10-17 FOR CHARACTER PUNCHED.
/
PTIT, TEXT -RANDOM SEQUENCE PUNCH ROUTINE.
     ACS0 DOWN FOR ERROR HALTS, UP FOR NONE.
     ACS1 DOWN FOR ERROR PRINT, UP FOR NONE.
     ACS17 DOWN TO START. PUT UP TO STOP PUNCHING.
-      
RTIT, TEXT -READER CHECK OF RANDOM TAPE.
     ACS0 AND ACS1 AS FOR PUNCH. ERRORS PRINTED BELOW.

CHARACTER PUNCHED EXPECTED

SOK, TEXT -DELAY OK. TURN OFF PUNCH POWER SWITCH.
-      
/ERROR MESSAGES
EM00, TEXT -STARTUP DELAY FAILURE. PRESS CONTINUE TO TRY IT AGAIN.
-      
EM01, TEXT -PSA 10 DID NOT CLEAR AC .
-      
EM10, TEXT -DID NOT CLEAR PUNCH FLAG.
-      
EM11, TEXT -PSF SKIPPED ON CLEARED FLAG.
EM13, TEXT - PSF FAILED TO SKIP WHEN FLAG WAS SET.

- EM20, TEXT - I WAITED 16 MSEC. BUT NO INTERRUPT. IS THE PUNCH REALLY ON?

- EM21, TEXT - I GOT AN INTERRUPT, BUT NOT FROM THE PUNCH. CHECK THE OTHER DEVICES. AC SHOWS I/O STATUS. PRESS CONTINUE TO START ALL OVER AGAIN.

START

/CONSTANTS, VARIABLES, ET ALIA

VARIABLES

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>COSP,</td>
<td>0</td>
</tr>
<tr>
<td>RAN1,</td>
<td>0</td>
</tr>
<tr>
<td>RAN2,</td>
<td>0</td>
</tr>
<tr>
<td>NONO,</td>
<td>0</td>
</tr>
<tr>
<td>APTEM,</td>
<td>0</td>
</tr>
<tr>
<td>EPT,</td>
<td>0</td>
</tr>
<tr>
<td>COSP/</td>
<td></td>
</tr>
</tbody>
</table>

/REVERSE FOR CONSTANTS, BELOW

73/

/RANDOM NUMBER PARTS

/RANDOM NUMBER COUNTERS

/ALL-PURPOSE TEMPORARY STORAGE

/FOR EM10 MNEMONIC

/HERE THERE BE CONSTANTS

PAUSE RAPT

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>71</td>
</tr>
<tr>
<td>APTEM</td>
<td>76</td>
</tr>
<tr>
<td>ATIT</td>
<td>402</td>
</tr>
<tr>
<td>B</td>
<td>104</td>
</tr>
<tr>
<td>BTATAB</td>
<td>1253</td>
</tr>
<tr>
<td>C</td>
<td>212</td>
</tr>
<tr>
<td>COSP</td>
<td>1363</td>
</tr>
<tr>
<td>COUNT</td>
<td>307</td>
</tr>
<tr>
<td>D</td>
<td>254</td>
</tr>
<tr>
<td>EM00</td>
<td>634</td>
</tr>
<tr>
<td>EM01</td>
<td>657</td>
</tr>
<tr>
<td>EM10</td>
<td>670</td>
</tr>
<tr>
<td>EM11</td>
<td>702</td>
</tr>
<tr>
<td>EM13</td>
<td>715</td>
</tr>
<tr>
<td>EM20</td>
<td>733</td>
</tr>
<tr>
<td>EM21</td>
<td>760</td>
</tr>
<tr>
<td>EPT</td>
<td>77</td>
</tr>
<tr>
<td>ER01</td>
<td>151</td>
</tr>
<tr>
<td>ER10</td>
<td>342</td>
</tr>
<tr>
<td>ER11</td>
<td>336</td>
</tr>
<tr>
<td>ER13</td>
<td>26</td>
</tr>
<tr>
<td>ER23</td>
<td>235</td>
</tr>
<tr>
<td>ER21</td>
<td>32</td>
</tr>
<tr>
<td>ER30</td>
<td>304</td>
</tr>
<tr>
<td>E00</td>
<td>64</td>
</tr>
<tr>
<td>E01</td>
<td>153</td>
</tr>
<tr>
<td>Code</td>
<td>Value</td>
</tr>
<tr>
<td>-------</td>
<td>-------</td>
</tr>
<tr>
<td>E10</td>
<td>216</td>
</tr>
<tr>
<td>E10P</td>
<td>347</td>
</tr>
<tr>
<td>E11</td>
<td>340</td>
</tr>
<tr>
<td>E12</td>
<td>227</td>
</tr>
<tr>
<td>E13</td>
<td>30</td>
</tr>
<tr>
<td>E20</td>
<td>240</td>
</tr>
<tr>
<td>E21</td>
<td>35</td>
</tr>
<tr>
<td>E30</td>
<td>325</td>
</tr>
<tr>
<td>E30R</td>
<td>322</td>
</tr>
<tr>
<td>E30P</td>
<td>313</td>
</tr>
<tr>
<td>F</td>
<td>311</td>
</tr>
<tr>
<td>FLAGT</td>
<td>327</td>
</tr>
<tr>
<td>FLOUT</td>
<td>345</td>
</tr>
<tr>
<td>ISR</td>
<td>23</td>
</tr>
<tr>
<td>NCT</td>
<td>1217</td>
</tr>
<tr>
<td>NONO</td>
<td>75</td>
</tr>
<tr>
<td>NON2</td>
<td>37</td>
</tr>
<tr>
<td>OCL</td>
<td>1232</td>
</tr>
<tr>
<td>OCN</td>
<td>1356</td>
</tr>
<tr>
<td>OCS</td>
<td>1233</td>
</tr>
<tr>
<td>OCO</td>
<td>1231</td>
</tr>
<tr>
<td>OPC</td>
<td>1355</td>
</tr>
<tr>
<td>OPR</td>
<td>1340</td>
</tr>
<tr>
<td>OPS</td>
<td>1317</td>
</tr>
<tr>
<td>OPT</td>
<td>1312</td>
</tr>
<tr>
<td>OPO</td>
<td>1332</td>
</tr>
<tr>
<td>OPI</td>
<td>1324</td>
</tr>
<tr>
<td>OP2</td>
<td>1346</td>
</tr>
<tr>
<td>OP3</td>
<td>1353</td>
</tr>
<tr>
<td>OTY</td>
<td>353</td>
</tr>
<tr>
<td>PACH</td>
<td>105</td>
</tr>
<tr>
<td>PACS</td>
<td>100</td>
</tr>
<tr>
<td>PALP</td>
<td>110</td>
</tr>
<tr>
<td>PAL2</td>
<td>121</td>
</tr>
<tr>
<td>PFEE</td>
<td>137</td>
</tr>
<tr>
<td>PFEL</td>
<td>141</td>
</tr>
<tr>
<td>PNEX</td>
<td>242</td>
</tr>
<tr>
<td>PRAN</td>
<td>223</td>
</tr>
<tr>
<td>PREC</td>
<td>251</td>
</tr>
<tr>
<td>PTIT</td>
<td>457</td>
</tr>
<tr>
<td>RACS</td>
<td>155</td>
</tr>
<tr>
<td>RAC2</td>
<td>166</td>
</tr>
<tr>
<td>RANDY</td>
<td>122</td>
</tr>
<tr>
<td>RAN1</td>
<td>73</td>
</tr>
<tr>
<td>RAN2</td>
<td>74</td>
</tr>
<tr>
<td>RAPT</td>
<td>200</td>
</tr>
<tr>
<td>RECH</td>
<td>274</td>
</tr>
<tr>
<td>RL6</td>
<td>1212</td>
</tr>
<tr>
<td>RTIT</td>
<td>545</td>
</tr>
<tr>
<td>SOK</td>
<td>616</td>
</tr>
</tbody>
</table>
11. DIAGRAMS (Not Applicable)

12. REFERENCES (Not Applicable)