THE NAS-PAK LOGIC SYSTEM

James Billingsley

For the past eight years, the design and fabrication of telemetry ground data processor equipment at Goddard Space Flight Center have been accomplished through the use of commercial logic cards or modules. Each design engineering group selected a manufacturer logic system which suited their particular requirements. Unfortunately, these logic systems were not physically or electrically compatible with each other; thus, competition was eliminated when additional purchases were made. As integrated circuits improved in performance and price, an opportunity for performance and size improvement motivated us to investigate new methods of implementation of the integrated circuits. A microelectronics program was initiated to evaluate circuitry, packaging methods, and fabrication approaches necessary to produce a competitively procured logic system. Goals of this microelectronic program were

(1) In integrated circuit evaluation, to evaluate circuit elements for their electrical performance, their availability from numerous sources, and the completeness of the logic line.

(2) In special circuit development, to design and test additional circuitry unavailable in the form of integrated circuits and to develop techniques that will allow competitive manufacture.

(3) In hardware development, to design and develop modular rack-mountable hardware to house the integrated circuits.

(4) In software development, to provide the designer with computer-generated design, fabrication, and drafting assistance to reduce cost and the incidence of human error.

(5) In testing and documentation, to evaluate available test equipment and fixtures for use with the new logic system and to provide a user's manual for the logic system and the program aids.

The result of this microelectronic program of investigation is the NAS-PAK logic system described in this report.

The NAS-PAK logic system emerged more from the integrated circuit dual inline package than from any other aspect of the logic system development. Circuit evaluation was performed without regard to package type. Hardware development, however, required consideration of each package type. The familiar printed circuit card was discarded because of connector pin number limitations. A socket panel with plug-in capacity for 60 14-pin dual inline packages was chosen. These panels (shown in Figure 1) are 15.2 cm X 17.8 cm X 0.32 cm with 14 wire-wrap pins per socket. Six socket panels are mounted on an aluminum frame (Figure 2) which in turn is mounted in an 8.9-cm drawer assembly (2 frames) or a vertical page assembly (3 frames). Figure 3 illustrates the modular assemblage from dual inline package to drawer assembly. Plastic protective covers are attached to the frame assemblies to prevent pin damage. The drawers assemblies have provision for mounting 4 fans, 2 fans, or a blank fan plate to facilitate cooling as required. The vertical page assembly is cooled by rack-mounted fans. Figure 4 illustrates the vertical page assembly (8 pages) mounted in a cabinet rack. The resultant reduction in size averages 15:1 over discrete component logic systems and 2:1 over commercial lines of integrated circuit modules.

Miniaturization of electronic equipment introduces the difficulty of circuit interconnection in very small areas. Automation of circuit interconnection then becomes necessary for full utilization of the system capabilities in normal time schedules. Realization of these problems early in the NAS-PAK system development led us to investigate wiring machines and their capabilities. The machine chosen, though manufactured by only one manufacturer, is available on a rental basis from several companies; thus, competition is preserved.

The wiring machine eliminated manual wiring errors and human deficiencies in the fabrication area but offered no improvement for design modification or correction after wiring. Computer programs were written to minimize errors introduced at the design phase. (See Figure 5.) These programs reduce the modifications and corrections required by eliminating the most common design errors before wiring. Figure 6 shows a typical intermediate printout demonstrating computer subroutines which provide design diagnostics, module placement guides, and circuit-use charts to further aid the user. Figure 7 illustrates the autowire program flow from logic diagram to the wired chassis. Additional programs were written to provide function block entry and automatic drafting of the designer's input data.

The block entry program reduces the input wirelist card number by generating wirelist cards from a single descriptive statement. Large shift registers and counters may be entered with a statement such as "–32 SRS." The block entry program will then generate all interconnection wiring cards necessary to wire the shift register as determined by the block entry function library. Since the designer does not have a drawing of the function, it became necessary to develop a program to show the designer how the computer implemented his input statement. The auto-draft program provides drive information to an *x-y* plotter to draw the logic symbols with module type, socket, and pin information. The symbols are positioned on the paper by coding information provided by the designer or the computer on the wirelist cards. The interconnecting lines are not drawn by this program and must be drawn by hand. Since the symbols and designation information consume 90 percent of hand drafting time, the auto-draft program offers considerable design assistance. Figures 8 and 9 are flow diagrams which show the complete NAS-PAK software system. User-oriented instruction manuals have been written for the NAS-PAK hardware and software.

The NAS-PAK logic system is a complete system covering all phases of implementation, not just the hardware phase alone. Each of the design aids were developed after the realization that previous developments introduced difficulties which would require additional effort by the designer. With these intentions, the NAS-PAK logic system will continue to develop.

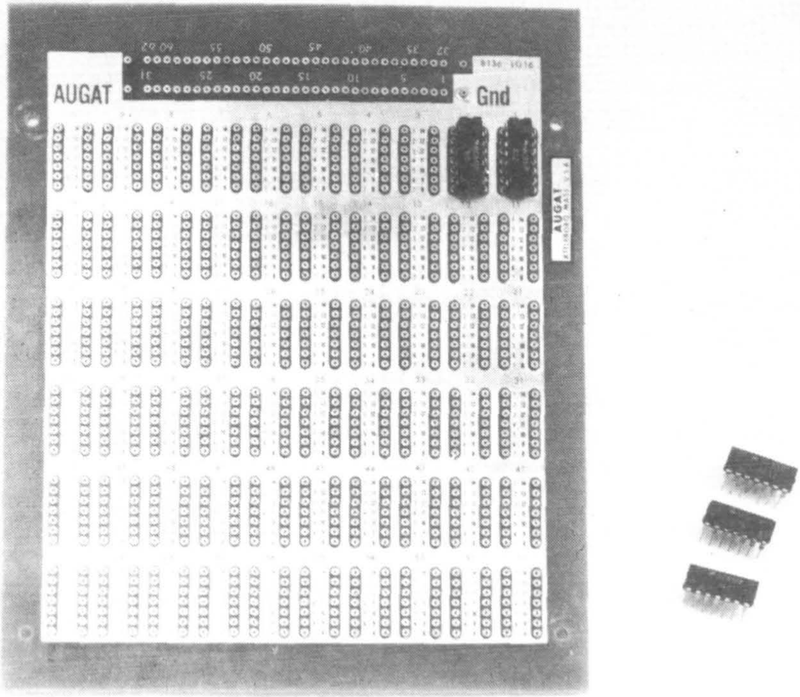


Figure 1–Socket panel.

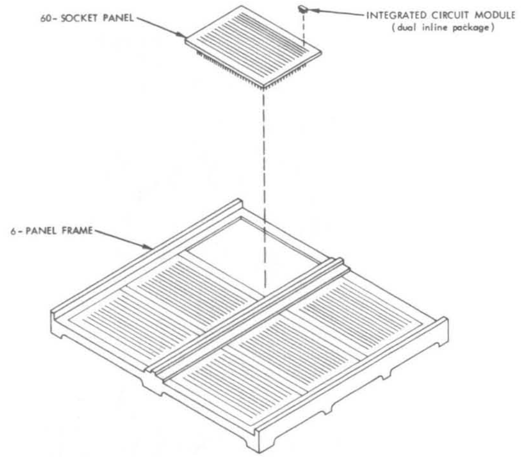


Figure 2–Socket panel mounting scheme.

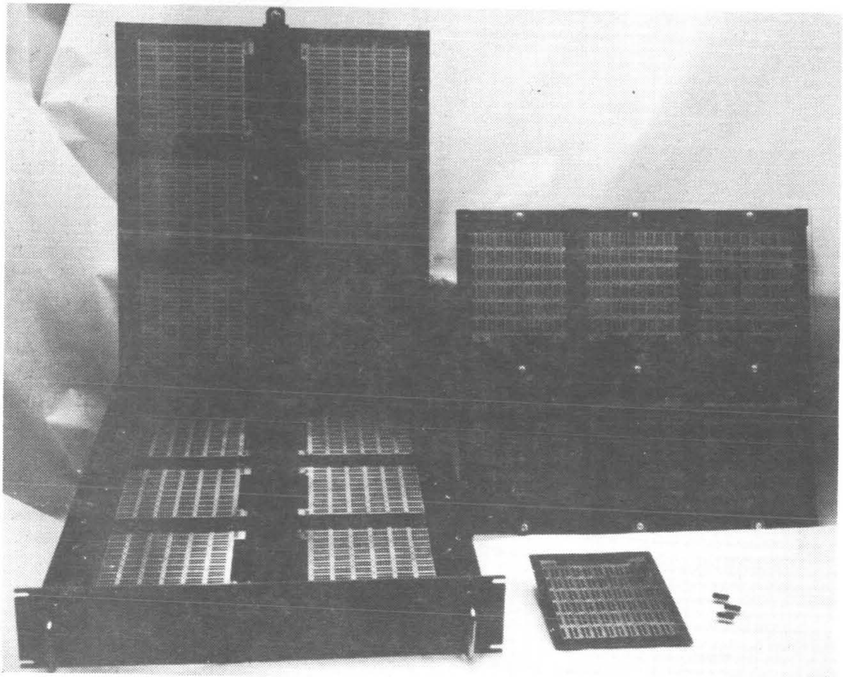


Figure 3–Modular assemblage.

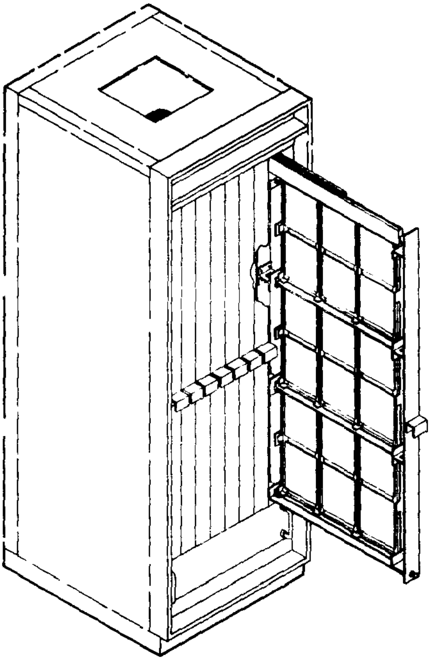


Figure 4–Vertical page assembly.

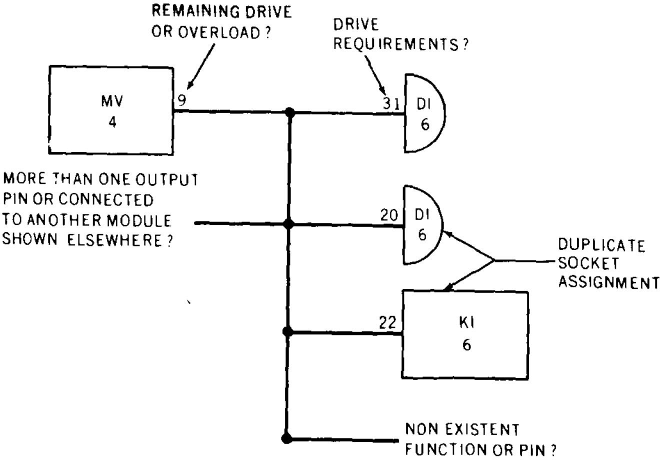
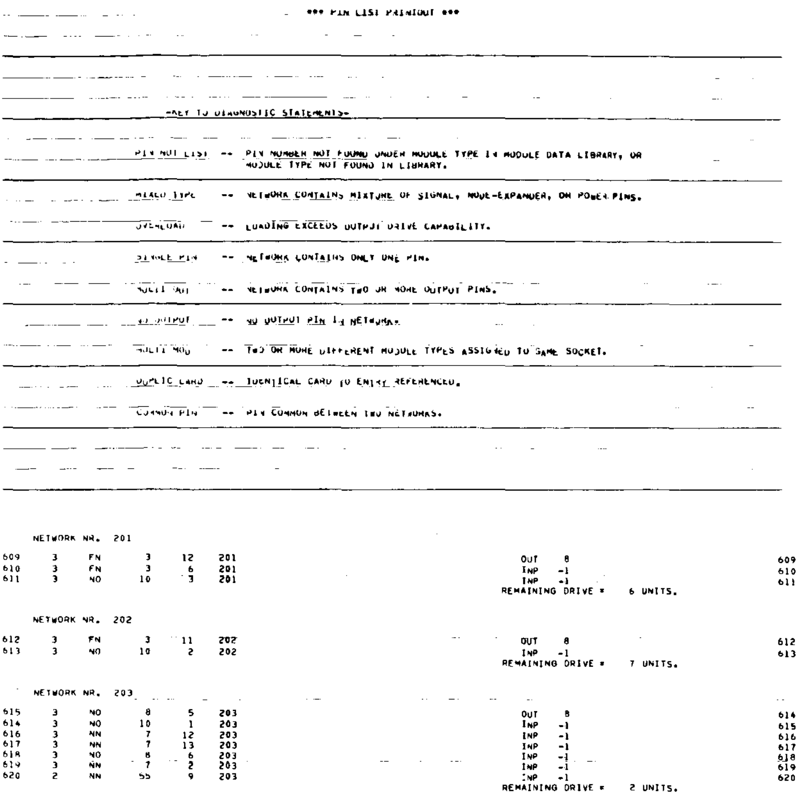
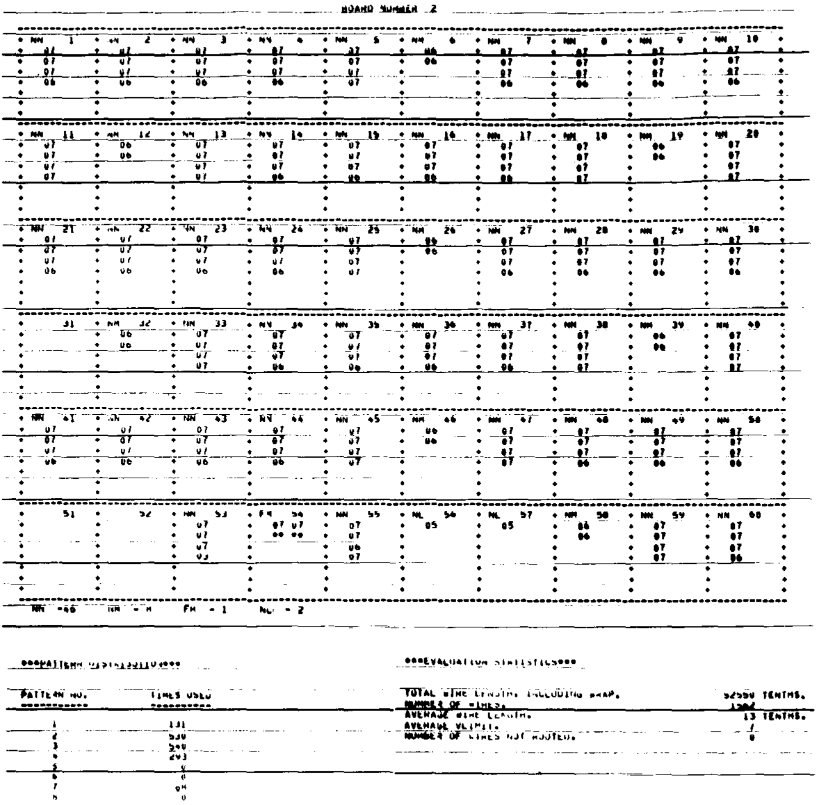

REMAINING DRIVE OR OVERLOAD ?
DRIVE REQUIREMENTS ?
MV
4
9
31
DI
6
MORE THAN ONE OUTPUT PIN OR CONNECTED TO ANOTHER MODULE SHOWN ELSEWHERE ?
20
DI
6
DUPLICATE SOCKET ASSIGNMENT
22
KI
6
NON EXISTENT FUNCTION OR PIN ?

Figure 5–Logic design aid program test.



```
                               *** PIN LIST PRINTOUT ***

                 -KEY TO DIAGNOSTIC STATEMENTS-

            PIN NOT LIST  --  PIN NUMBER NOT FOUND UNDER MODULE TYPE IN MODULE DATA LIBRARY, OR
                              MODULE TYPE NOT FOUND IN LIBRARY.

            MIXED TYPE    --  NETWORK CONTAINS MIXTURE OF SIGNAL, NODE-EXPANDER, OR POWER PINS.

            OVERLOAD      --  LOADING EXCEEDS OUTPUT DRIVE CAPABILITY.

            SINGLE PIN    --  NETWORK CONTAINS ONLY ONE PIN.

            MULTI OUT     --  NETWORK CONTAINS TWO OR MORE OUTPUT PINS.

            NO OUTPUT     --  NO OUTPUT PIN IN NETWORK.

            MULTI MOD     --  TWO OR MORE DIFFERENT MODULE TYPES ASSIGNED TO SAME SOCKET.

            DUPLIC CARD   --  IDENTICAL CARD TO ENTRY REFERENCED.

            COMMON PIN    --  PIN COMMON BETWEEN TWO NETWORKS.


     NETWORK NR.  201

609     3     FN      3    12    201                             OUT    8                        609
610     3     FN      3     6    201                             INP   -1                        610
611     3     NO     10     3    201                             INP   -1                        611
                                                          REMAINING DRIVE =    6 UNITS.

     NETWORK NR.  202

612     3     FN      3    11    202                             OUT    8                        612
613     3     NO     10     2    202                             INP   -1                        613
                                                          REMAINING DRIVE =    7 UNITS.

     NETWORK NR.  203

615     3     NO      8     5    203                             OUT    8                        614
614     3     NO     10     1    203                             INP   -1                        615
616     3     NN      7    12    203                             INP   -1                        616
617     3     NN      7    13    203                             INP   -1                        617
618     3     NO      8     6    203                             INP   -1                        618
619     3     NN      7     2    203                             INP   -1                        619
620     2     NN     55     9    203                             INP   -1                        620
                                                          REMAINING DRIVE =    2 UNITS.
```

Figure 6—Computer subroutines.



```
                                        BOARD NUMBER  2

*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
* NN    1 * NN    2 * NN    3 * NN    4 * NN    5 * NN    6 * NN    7 * NN    8 * NN    9 * NN   10 *
*   07    *   07    *   07    *   07    *   07    *   06    *   07    *   07    *   07    *   07    *
*   07    *   07    *   07    *   07    *   07    *   06    *   07    *   07    *   07    *   07    *
*   07    *   07    *   07    *   07    *   07    *         *   07    *   07    *   07    *   07    *
*   06    *   06    *   06    *   06    *   07    *         *   06    *   06    *   06    *   06    *
*         *         *         *         *         *         *         *         *         *         *
*         *         *         *         *         *         *         *         *         *         *
*         *         *         *         *         *         *         *         *         *         *
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
* NN   11 * NM   12 * NN   13 * NV   14 * NN   15 * NN   16 * NN   17 * NN   18 * NN   19 * NN   20 *
*   07    *   06    *   07    *   07    *   07    *   07    *   07    *   07    *   06    *   07    *
*   07    *   06    *   07    *   07    *   07    *   07    *   07    *   07    *   06    *   07    *
*   07    *         *   07    *   07    *   07    *   07    *   07    *   07    *         *   07    *
*   07    *         *   07    *   06    *   06    *   06    *   06    *   07    *         *   07    *
*         *         *         *         *         *         *         *         *         *         *
*         *         *         *         *         *         *         *         *         *         *
*         *         *         *         *         *         *         *         *         *         *
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
* NN   21 * NN   22 * NN   23 * NN   24 * NN   25 * NM   26 * NN   27 * NN   28 * NN   29 * NN   30 *
*   07    *   07    *   07    *   07    *   07    *   06    *   07    *   07    *   07    *   07    *
*   07    *   07    *   07    *   07    *   07    *   06    *   07    *   07    *   07    *   07    *
*   07    *   07    *   07    *   07    *   07    *         *   07    *   07    *   07    *   07    *
*   06    *   06    *   06    *   06    *   07    *         *   06    *   06    *   06    *   06    *
*         *         *         *         *         *         *         *         *         *         *
*         *         *         *         *         *         *         *         *         *         *
*         *         *         *         *         *         *         *         *         *         *
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
*      31 * NM   32 * NN   33 * NV   34 * NN   35 * NN   36 * NN   37 * NN   38 * NM   39 * NN   40 *
*         *   06    *   07    *   07    *   07    *   07    *   07    *   07    *   06    *   07    *
*         *   06    *   07    *   07    *   07    *   07    *   07    *   07    *   06    *   07    *
*         *         *   07    *   07    *   07    *   07    *   07    *   07    *         *   07    *
*         *         *   07    *   06    *   06    *   06    *   06    *   07    *         *   07    *
*         *         *         *         *         *         *         *         *         *         *
*         *         *         *         *         *         *         *         *         *         *
*         *         *         *         *         *         *         *         *         *         *
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
* NN   41 * NN   42 * NN   43 * NV   44 * NN   45 * NM   46 * NN   47 * NN   48 * NN   49 * NN   50 *
*   07    *   07    *   07    *   07    *   07    *   06    *   07    *   07    *   07    *   07    *
*   07    *   07    *   07    *   07    *   07    *   06    *   07    *   07    *   07    *   07    *
*   07    *   07    *   07    *   07    *   07    *         *   07    *   07    *   07    *   07    *
*   06    *   06    *   06    *   06    *   07    *         *   07    *   06    *   06    *   06    *
*         *         *         *         *         *         *         *         *         *         *
*         *         *         *         *         *         *         *         *         *         *
*         *         *         *         *         *         *         *         *         *         *
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
*      51 *      52 * NN   53 * FN   54 * NN   55 * NL   56 * NL   57 * NM   58 * NN   59 * NN   60 *
*         *         *   07    *  07 07  *   07    *   05    *   05    *   06    *   07    *   07    *
*         *         *   07    *  06 06  *   07    *         *         *   06    *   07    *   07    *
*         *         *   07    *         *   06    *         *         *         *   07    *   07    *
*         *         *   03    *         *   07    *         *         *         *   07    *   06    *
*         *         *         *         *         *         *         *         *         *         *
*         *         *         *         *         *         *         *         *         *         *
*         *         *         *         *         *         *         *         *         *         *
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
  NN =46     NM = 8     FN = 1     NL = 2


***PATTERN DISTRIBUTION***                        ***EVALUATION STATISTICS***

PATTERN NO.        TIMES USED                     TOTAL WIRE LENGTH, INCLUDING WRAP.            52550 TENTHS.
-----------        ----------                     NUMBER OF WIRES.                               1592
                                                  AVERAGE WIRE LENGTH.                             33 TENTHS.
     1                131                         AVERAGE HEIGHT.                                   1
     2                530                         NUMBER OF WIRES NOT ROUTED.                       0
     3                540
     4                293
     5                  0
     6                  0
     7                 98
     8                  0
```

Figure 6–Computer subroutines (concluded).

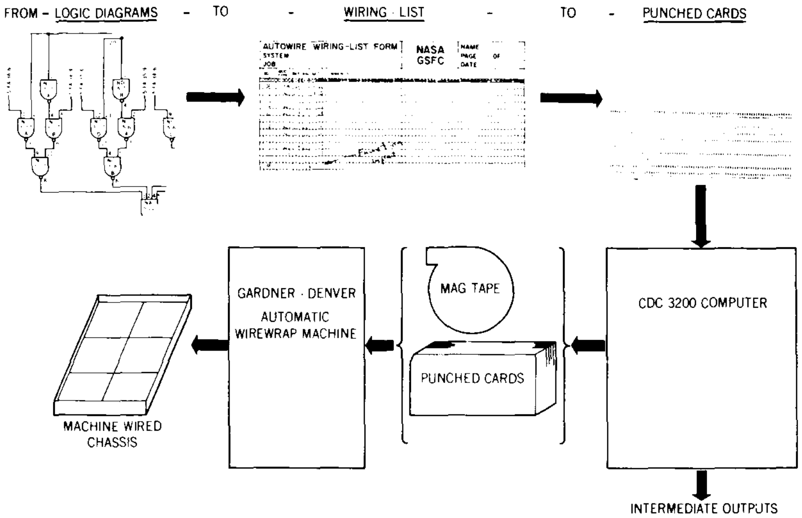

FROM - LOGIC DIAGRAMS - TO - WIRING - LIST - TO - PUNCHED CARDS
AUTOWIRE WIRING-LIST FORM
NASA GSFC
CDC 3200 COMPUTER
MAG TAPE
PUNCHED CARDS
GARDNER - DENVER AUTOMATIC WIREWRAP MACHINE
MACHINE WIRED CHASSIS
INTERMEDIATE OUTPUTS

Figure 7—Autowire program flow.

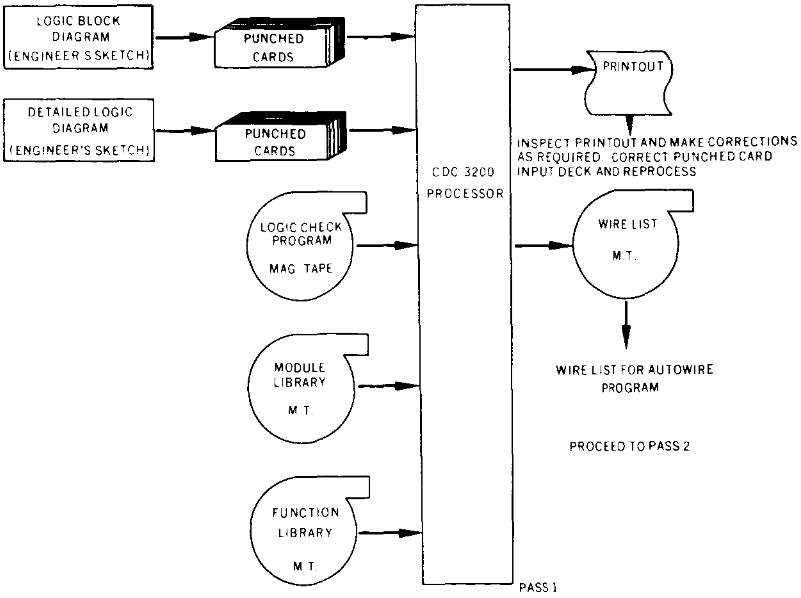

LOGIC BLOCK DIAGRAM (ENGINEER'S SKETCH)
PUNCHED CARDS
DETAILED LOGIC DIAGRAM (ENGINEER'S SKETCH)
PUNCHED CARDS
LOGIC CHECK PROGRAM MAG TAPE
MODULE LIBRARY M.T.
FUNCTION LIBRARY M.T.
CDC 3200 PROCESSOR
PRINTOUT
INSPECT PRINTOUT AND MAKE CORRECTIONS AS REQUIRED. CORRECT PUNCHED CARD INPUT DECK AND REPROCESS
WIRE LIST M.T.
WIRE LIST FOR AUTOWIRE PROGRAM
PROCEED TO PASS 2
PASS 1

Figure 8—Logic design aid program.

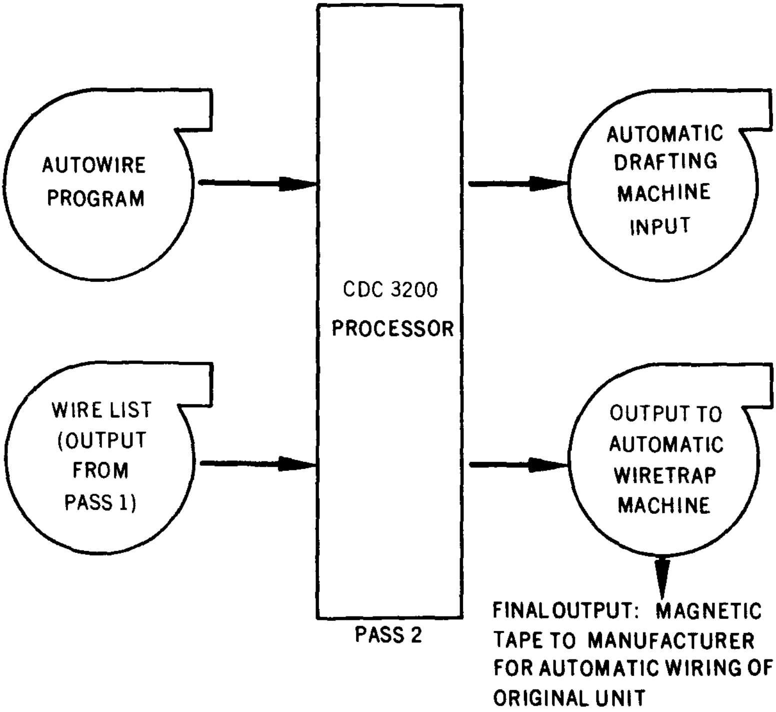

AUTOWIRE PROGRAM
WIRE LIST (OUTPUT FROM PASS 1)
CDC 3200 PROCESSOR
PASS 2
AUTOMATIC DRAFTING MACHINE INPUT
OUTPUT TO AUTOMATIC WIRETRAP MACHINE
FINAL OUTPUT: MAGNETIC TAPE TO MANUFACTURER FOR AUTOMATIC WIRING OF ORIGINAL UNIT

Figure 9—Automatic wiring program.