(specification)

CAD MEMO 3

# MICROCOMPUTER WRAPB PROGRAM DISK

S. Keller
March 7, 1986

## I. INTRODUCTION

This report describes a microcomputer version of a program designed to minimize the work involved in producing a wiring list for NRAO Mark III wirewrap/printed-circuit boards (shown in Figure 1). The program is intended to replace the FORTRAN programs known as WrapB and Convert for the IBM mainframe ("Pandora"). That program, and the one described here, were written as tools to let a logic designer take a logic diagram and translate it into a wiring list for a Mark III wirewrap card. This new version has greater flexibility in chip placement and pin specification, better error trapping and output format, and outputs both used and unused pin cross-reference lists. Single Inline Packages can be entered, as well as redefining the Vcc and GND pins for non-standard packages. In addition, descriptions are provided for related programs to prepare the input file, and to process the output files for conversion to other formats. It is assumed that the reader is familiar with the internal report describing the microcomputer version of WireWrap, used for the NRAO Shalloway wirewrap boards.

## II. WRAPB DISK DESCRIPTION

The program disk is a stand-alone IBM-PC compatible PC-DOS/MS-DOS 5-1/4" floppy disk containing several files, including the PC-WRITE text editor (and its associated files), a brief help file, two versions of the wirewrap program (one for use those machines with an 8087 math chip installed, the other for non-8087 machines), and a program to convert the wirewrap output file to a compressed format for use by an automatic wirewrap company such as ELMA. A batch file is used to automate the wirewrap processing sequence; it is designed to catch errors and abort gracefully whenever possible. Due to memory constraints, the source code for most of the programs, and the Turbo Pascal compiler under which it was developed, are actually stored on a separate disk; in general, when WrapB or the WrapB disk is mentioned, it refers to the dedicated WrapB disk with the executable files, not the source code.

The PC-WRITE text editor is provided on the WrapB disk for entering the wirelist input file; if the user wishes, any editor that produces DOS text files may be used instead. It is recommended that the user start with a blank formatted disk and maintain all his files on that disk instead of using the wirewrap disk. The wirelist file should contain the chip declarations and wirelist information in the format described later in this report.

After saving this file to his work disk, the user may then boot the wirewrap program disk (or simply invoke AUTOEXEC.BAT); a title screen will identify the disk and offer suggestions on the available programs. By entering "WB" (without quotes) at the DOS prompt, a batch file will execute and invoke the wirewrap program. The user then specifies the name of the wirelist input and output files. If no errors occur, there will normally be a wire list output file and a pin cross-reference file on disk, and the batch file will resume execution and ask the user if he wants to proceed to the conversion program to generate an ELMA-compatible (or other) wirelist file.

## III. USING THE WRAPB PROGRAM

WrapB is designed to work on the IBM PC and compatibles (including the AT&T) with DOS 2.0 or higher, and the format of the input and output files of

WrapB are normal PC-DOS/MS-DOS ASCII text files. The computer should have at least 256K of memory in order to run WrapB; 640K is required if a large RAM disk is going to be used with the program. To facilitate the generation of the input file, the PC-WRITE text editor is provided on the disk, but almost any other editor that generates a text file can be used.

The simplest way to use WrapB is to boot the disk, and after the title page is displayed, invoke the PC-WRITE editor with "ED filename", where 'filename' is the name of the input file you wish to create. Working from the original schematics and circuit board layout, you are expected to enter a file in the format described in Section IV.

After you have entered the input file, save it to disk and exit the editor. From the DOS prompt, type 'WB' to start the wirewrap process. WB is a batch file that will determine if an 8087 numeric coprocessor is installed in your machine, and will invoke the appropriate version (8087 or non-8087) of WrapB. WrapB will display a few title lines, and then ask for an input filename, which may include a drive specification; if the file cannot be successfully opened, you will be prompted again. You will then be asked for the name of the output file for the wirelist and an output file for the pin cross-reference. In both these cases you may include a drive specification, and if a file cannot be successfully created, the program will prompt you for filenames until it is successful. It is also possible to specify 'CON:' or 'NUL' as a filename. Using CON: will send the output file to the screen (console) instead of the disk; if an output file is not desired, NUL can be used to act as a bit bucket by throwing away any data it receives. CON: and NUL are most commonly used during the debugging stages to quickly display errors in the wirelist, and for suppressing the generation of a cross-reference file, which can spare the user from several minutes of execution time.

If the wirelist output is sent to disk, WrapB will print a period (".") to the screen for every wirelist string that is output, reassuring the user that the program is still functioning. (The length of the pauses between the printing of periods is directly related to the number of pins, and hence the amount of processing, in the current wire string.)

Almost all errors that can occur are sent to the wirelist output file, and are placed on lines by themselves, prefixed with "!", to make them easier to locate. WrapB was designed to continue processing after errors occur to allow locating as many as possible in one run of the program. Be forewarned, though, that some errors can easily cascade into others: for example, a bad IC declaration will almost certainly cause bad chip references in the wirelist. To avoid overflowing a disk with error messages, though, WrapB is designed to stop at 25 errors.

Due to the more complex nature of the Mark III board, execution times for WrapB generally takes longer than a comparably sized board would under WireWrap. Using a RAM disk or hard disk will substantially reduce execution time; files can always be copied to another disk at a later time. If a cross-reference file is generated, it will contain an alphanumeric list of used pins with the input file line number, and the one or two other pins to which it is connected. (Both IC pins and backplane notation pins will be output in this list.) Following that list is an alphanumeric list of unused IC pins (but no backplane pins, since such a list would be prohibitively long).

If errors occurred during WrapB, the user will be notified on the screen with the error count, and both WrapB and the batch file will terminate. If there were no wirelist errors, the batch file resumes control and will give the user an opportunity to proceed to a conversion program to convert the output file to ELMA format for automatic wirewrapping. It is intended that this conversion program will support new formats in the future if that should be required.

To view the output files, any one of several methods can be used: PC-WRITE, the DOS TYPE command, the text file list utility "L", or the DOS COPY command, which can send the output files to the printer ("PRN:"). All of these are available on the WrapB disk.

## IV. INPUT FILE FORMAT

The input text file is expected to have the following general format:

```
* Comment lines: any text that occurs after an asterisk in a line
*  is ignored.  This allows commenting the file with date and project
*  name, as well as with details regarding each chip declaration or
*  other useful information regarding the wirelist commands.
SECTIONS=6
* The command above informs WrapB how many sections there are in
*  the Mark III board being used; legal values are 1 through 6,
*  representing the sections A through B.
DECLARE
* The command above, which must appear on a line by itself, informs
*  WrapB to begin processing IC declarations, one to a line, in
*  the format below:
* Loc'n    # Pins   Pin #1 Loc'n   Type     Vcc    GND
A11           20          A11         1        20     1  * (EXAMPLE)
* Note: there must be at least one space separating each column;
*  additional spaces between columns are ignored, and can be used
*  to enhance readability. See the text for a discussion of the
*  information specified in IC declarations.
```

```
WIRELIST
* The command above informs WrapB that the chip declarations are
*  complete, and that the list of wiring instructions follow.
S A1101 AJ137 @A1121     * (Example wire string specification)
* Wire strings are sequences of two or more pins to be wired together;
*  the two string commands are 'S' and 'SE' (for 'String' and 'String
*  and 'String Exact').  See the text for a discussion of pin
*  specifications, wire string limitations, etc.
END
* The command END is used to mark the end of the entire input file; any
*  additional lines will be ignored.
```

IC declarations must contain the following information:

```
* Loc'n     # Pins    Pin #1 Loc'n    Type     Vcc     GND
A11           20         A1101          1       20       1  * (EXAMPLE)
```

Name : Frontplane name of the IC, specified as a section letter A through F, followed by a 2-digit pattern number. In the event that more than one package exists in one pattern, a letter may be appended to the chip name to distinguish it from other chips in that pattern. (E.G., "A23A","A23B", etc.)

# Pins : Total number of pins on IC or package

Pin #1
Loc'n : Location, in boardplane notation, of pin #1 on the IC. This consists of a section letter (A..F), followed by a pattern number, and then a board pin number 01 through 20. Note that the location is usually, but not necessarily, the same as the chip name; this allows for convenient naming of 0.6" chips, which must be placed across section boundaries.

Type : Specifies the size of the chip in multiples of 0.3 inches: 1 means a .3 inch chip, 2 means .6 inch, etc. For Single Inline Packages, use 0.

>>>>NOTE: SPECIFICATION OF VCC AND GND ARE OPTIONAL IN A RESTRICTED WAY: YOU MAY OMIT BOTH OR INCLUDE BOTH, BUT YOU MAY NOT SPECIFY ONLY ONE. FOR THOSE CIRCUMSTANCES IN WHICH A VCC OR GND PIN DO NOT APPLY, SUCH AS A RESISTOR PACK, USE A "0" FOR THAT PIN.<<<<<

Vcc : Pin number on IC of Vcc. If not specified, this value defaults to the highest numbered pin. Zero means there is no Vcc pin.

GND : Pin number of GND. If not specified, this defaults to the value (highest numbered pin) / 2. Zero means there is no GND pin.

Also note that single digit values are normally converted to 2-digit values (prefixed with zero) when appropriate, and will appear as such in the output files.

In the wirelist section of the file, there are two ways to wire up strings: "S" (for "String") specifies all pins in the string and lets WrapB determine the optimal sequence in which to wire those pins so as to minimize the total length of wire used. The other command, "SE" (for "String Exact") specifies an exact sequence in which pins are to be wired. Up to 50 pins are allowed in a string; a single line in the input file may contain one or more wire strings, or a single wire string may cross two or more lines. In general, wiring will continue until an "S" or "SE" is encountered. In both cases, the order of the wires will be staggered to make repairs easier, but the ultimate sequence will be the same. If bad IC or pin references occur, there may not be enough pins to form a string, and WrapB will generate a corresponding error message.

A pin specification may be either a frontplane coordinate (also called chip notation, since it refers to a pin number on a particular chip or package), or a boardplane coordinate prefixed with "@", or a prewired Vcc, GND, or VX pin. ("VX" refers to pin 21 of each pattern, which is pre-wired to either V1 or V2 on the board.) Such pins are specified by using "VC", "GD", or "VX" (respectively) as the last two characters of the pin specification (e.g., "@A11VC"). Frontplane coordinates normally have the form of a section letter (A through F), a two-digit pattern number, an optional letter for distinguishing between multiple packages in the same pattern, and a two-digit pin number. For example, "A1102" refers to IC A11, pin number 2; "B23A01" refers to IC "B23A", pin 1. Boardplane coordinates have a similar format (which caused no end of confusion for this author); they are formed with an "@" followed by a section letter A..F, followed by a two-digit pattern number and then a two-digit pin number, 1..21. There are some exceptions to this format: the special columns 10,17,20,27,..60,67 have only ten pins, and are most commonly used for positioning 0.6" chips; another exception is the jumper block, which consist of four rows of 25 pins each. The first two rows are called Jumper 1, and the next two rows Jumper 2, with their pins numbered from 1 through 50. Jumper pins are normally entered in the format "AJ136", in this case meaning section A, Jumper block 1, pin 36.

Unlike WireWrap, which performs automatic Vcc and GND wirings, the user is responsible for entering all such connections when using WrapB.

## V. OUTPUT FILE FORMATS

WrapB can produce two files: a wirelist output file and a used/unused pin cross-reference list. The wirelist output file is a DOS text file which contains the following sections of information:

-Wirelist. This is the format:

Example:

```
  INPUT LINE    LENGTH   FROM      TO        COUNT
 Start
           20       3.0     @A1101    @A1102       1      A1101   A1102
```

Length is the wire length in inches (see note on wire length elsewhere). "FROM" and "TO" are the boardplane coordinates of the wire to be connected, and "COUNT" is the cumulative wire count for the wirelist. Each wire string begins with the word "Start", followed by one or more lines containing each pair of pins to be wired together. Also, the "INPUT LINE" column contains the input line number from the WrapB input text file, and the last two items on the line are the original pin specifications from the input file. (Having these pin specifications displayed here is extremely useful in the debugging process). Note that wire length is calculated as the Pythagorean distance between the two pins, plus 2.5 inches to allow slack during wirewrapping, rounded up to the nearest half inch. If this length is less than 3 inches, it is set equal to 3 inches, to insure that there is enough wire to allow wrapping.

-The last line of a wirelist output file should be "END OF WIRELIST"; if this is not the case, then the file is probably missing the last part of its output (perhaps due to a disk error).

The pin cross-reference file is the other output file that WrapB can generate. It consists of:

-Used pin cross-reference, in alphanumeric order of pins. Both frontplane and boardplane notation pins will appear. The first column contains the input file line number, then the pin currently being cross-referenced, and the one or two pins to which it is connected.

-Unused pin cross-reference (for IC pins only) in alphanumeric order. A carriage return is generated after each chip and/or after every eight pins output from a single chip. The last line of this file is "END OF XREF FILE".

## VI. PROGRAM DESCRIPTION

WrapB is written in Turbo Pascal v3.01A for MS-DOS/PC-DOS 2.0 or higher. Both the regular (non-8087) and 8087 Turbo Pascal compilers were used to generate the two versions of the wirewrap program on the disk. The source code for WrapB is approximately 50K long and generates a .COM file about 30K long. A variety of static and dynamic data structures are used to store the various data; constants are used where possible, and many variables are represented by structured variables. WrapB was derived from the PC version of WireWrap, and is designed to function as similarly as possible to that program for those users who must deal with both types of wirewrap boards, as well as to facilitate future program maintenance.

Several goals were kept in mind when WrapB was written: it needed to replace the mainframe version of the WrapB program, but include features to handle situations that were not accounted for when the mainframe version was written. Source file formats had to be similar to allow easy conversion of mainframe files to the new program, but allow for flexible placement of chips and declaration of chips to handle different size packages, the ability to override the defaults for the Vcc and GND pin numbers, and easier specification of boardplane pin locations. In addition, a new format of pin cross-reference was desired along with an unused IC pin cross-reference. Any errors that occur during processing should not terminate WrapB but rather give an appropriate error message, increment an error count and continue processing.

Rather than port the original program to the PC, it was decided to rewrite the entire program using a language and programming environment that would allow for easier program maintenance and future modification, as well as providing reasonable execution speed. Most of the program structure had already been written for the PC WireWrap program, so it was decided to modify that program to handle the Mark III boards. Much of the PC WireWrap report also applies to the WrapB program description, and the reader is referred to that report, while this report will provide only important differences between the two programs.

Unlike WireWrap, WrapB does not automatically generate Vcc and GND pin wiring, nor does it provide a bypass capacitor list. Most of the WireWrap program had to be re-written to handle the different frontplane and boardplane notation that is used with the Mark III boards. Determining the legality of boardplane pins is far more complicated with the Mark III board, as is the translation between frontplane and boardplane coordinates. Three arrays are used to keep track of pin usage in WrapB (one for jumper pins, one for pin 21 of each pattern, and one for the regular pins), whereas WireWrap uses only one array. Functions were written for storing and retrieving the pin usage value to/from the appropriate array, given the boardplane coordinate. The maximum number of chips and pins in a string list was increased; functions and procedures were written to handle a "SECTIONS=" declaration at the beginning of the input file, to check for the legality of pin specifications, and many other routines from WireWrap were modified to handle the peculiarities of the Mark III board. For example, much effort was spent to handle 0.6" chips, which must be placed across section boundaries, and consequently require special processing to convert their pins to boardplane coordinates.

The optimal string and exact string procedures have not been changed; the procedure to print the unused pin list has been changed to handle the different syntax of frontplane coordinates.

## VII. ADDITIONAL PROGRAMS

The WrapB disk has several files on it in addition to the WrapB program. There is a DOS batch file called 'WB' that is used in automating the wirewrap process; it makes use of several batch file commands in determining if desired files exist, as well as examining the 'ErrorLevel' upon exit of certain programs to determine flow of control in the batch file. For example, WB calls a .COM file to test for the presence of an 8087 numeric coprocessor. Based on the results of that test, WB will invoke the regular (non-8087) or 8087 version of WrapB. After exiting WrapB, an ErrorLevel of 0 indicates that no errors occurred, and WB will offer the user the opportunity to proceed to the ELMA conversion program. The WB batch file is presented in Listing 1.

The TEST8087 file is a machine language file for detecting the presence or absence of an 8087 numeric coprocessor in the system, and returns that information to the WB batch file described above. The program consists of two parts: a short Turbo Pascal program, and a machine language program. The machine language program was modified from a utility that was published in the June, 1985, PC TECH JOURNAL magazine, p. 181; it performs the actual test to see if the 8087 is installed, and passes the information back to the Turbo Pascal program, which displays a message to the user, and then passes the information to the batch file via the "Halt(0)" or "Halt(1)" commands. Listing 2 contains the two parts of the TEST8087 program.

The ELMA convert program (CONVWB) is a short (140 line) Turbo Pascal program that will take the wirelist output file from WrapB and convert it into a new file in the format required for ELMA's automatic wirewrap equipment. The output file to be processed must not have any errors, or CONVWB will abort the conversion process. The user is asked for the names of the file to be processed and the filename for storing the converted information. In addition, the user will be asked for a one line text header to place at the beginning of the converted file. CONVWB will read lines from the file to be converted, ignoring any lines that do not contain wiring information (such as the column headers and 'Start' lines). The two pins to be wired and the wiring count will be extracted and output to the new file in the rigid ELMA format. Note that CONVWB expects such information to be in specific places in WrapB's output file; any substantial change to that format will have to be accounted for in the CONVWB program.

A separate program which has proved useful in transferring old wirewrap input files from the IBM mainframe to the PC is COPYFILE, which is a short Pascal program on the PC for eliminating undesirable control characters, trailing spaces, and optionally removing other unwanted lines from files that have been downloaded. For example, some downloaded files are prefixed with line numbers, and/or contain blank lines and lines of asterisks for formatting purposes. COPYFILE gives the user the option to remove some or all of these groups of characters. Substantial amounts of disk space can be saved by processing files with this utility, as well as simplifying the conversion of old wirewrap source files to the new format.

## VIII. CONCLUSION

This report has described the personal computer version of WrapB, a program to facilitate the generation of wirewrap wiring lists from the original circuit schematics and IC placement diagrams. WrapB was written to replace the wirewrap program on the IBM mainframe, as well as to offer capabilities and features not available in the mainframe version. These new features include more flexible chip placement, handling of different IC sizes including Single Inline Packages, more convenient backplane notation and better error trapping. Additional programs with WrapB include a conversion program of WrapB output to ELMA compatible file format, and a filter program that is useful for converting files downloaded from the mainframe into a DOS text file, with all control characters (other than carriage returns) and trailing spaces removed.

## IX. ACKNOWLEDGMENTS

This project was done under the direction of Ray Escoffier, Walter Brown, Gene Runion and Andy Dowd, and the author would like to thank these people for their help.

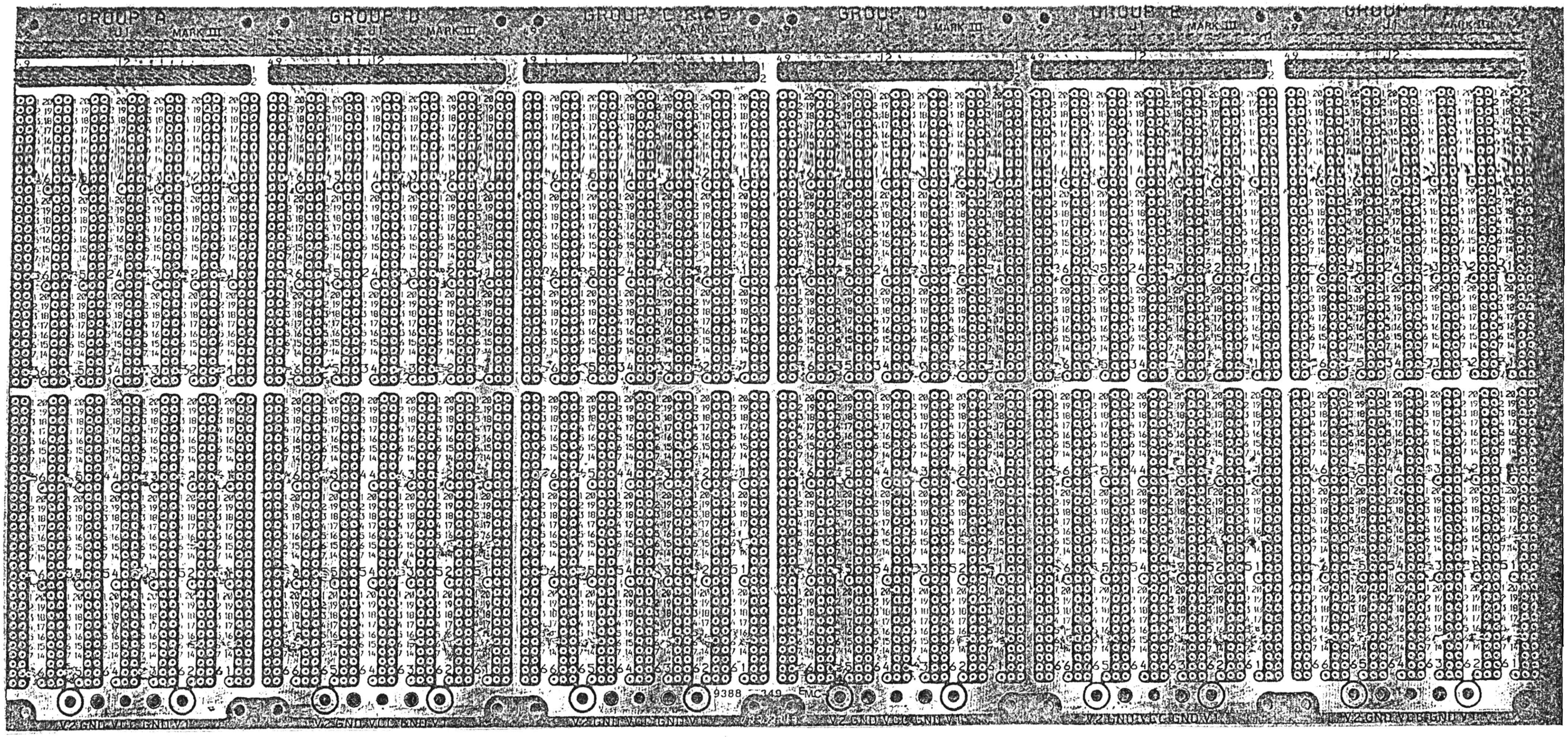

Figure 1. Mark III card.

## Listing 1. The WB Batch File

```
echo off
echo testing for 8087 math chip
test8087
if errorlevel 1 goto wb

if not exist wrapb87.com goto fileerr
echo To invoke the 8087 WrapB program,
pause
wrapb87
if errorlevel 1 goto exit
goto convert

:wb
if not exist wrapb.com goto fileerr
echo To invoke the regular WrapB program,
pause
wrapb
if errorlevel 1 goto exit
goto convert

:fileerr
echo ERROR: CAN'T FIND THAT FILE OR PROGRAM
goto exit
:convert
echo To convert output to ELMA format,
pause
convwb

:exit
echo termination of batch file.
```

## Listing 2. Listing of TEST8087 Programs

```
program testfor8087;    {Turbo Pascal program}

function test8087 : integer; external 'test'; {use machine language test}

begin {program}
  if test8087 = 0 then
    begin
      Writeln('False--8087 or 80287 not installed.');
      Halt(1);  {exit with errorlevel = 1}
    end
  else
    begin
      Writeln('True--8087 or 80287 installed.');
      Halt(0);   {exit with errorlevel = 0}
    end;
end. {program}
```

```
; Assembly language utility to detect 8087:
; Check for 8087 or 80287 math coprocessor
; modified from june 1985 pc tech journal, p. 181

code    segment public
assume cs:code

        org     100h                    ;set up as a com file
        .8087                           ;tell assembler to process 8087 stuff
start:  jmp     begin

control dw      0

;
; test if math coprocessor is present and return integer (0 or 1) to
;    the turbo pascal program

begin:
        fninit                          ;initialize math chip
        xor     ah,ah                   ;clear ah
        mov     byte ptr CS:control+1,ah ;clear memory byte
        fnstcw  control                 ;store control word
        mov     ah,byte ptr CS:control+1 ;hi byte is 03h if 8087
        cmp     ah,03h                  ;   or 80287 installed
        jne     none                    ;
        mov     ax,01h                  ;math chip is present
        jmp     exit                    ;
none:
        mov     ax,00h                  ;math chip not present
exit:
        ret                             ;return
code    ends                            ;
        end     start                   ;start is entry point
```