DataBoard 4680 PROM PROGRAMMER provides low cost and highly versatile solution on all your PROM-programming required when running up an application based on the DataBoard 4680 product line. The different cases, where PROM:s are required are:

1. PROM-modules for your programs and writeprotected data.

   The PROM-modules are:
   3061 using INTEL 2758, 2716 or 2732 or equivalent types
   3086 using Texas Instrument 2532 or equivalent

2. On-Board PROM-programs in Single Board Computers:

   Using INTEL 2758 or equivalent
   or   INTEL 2716 "   "
   or   INTEL 2732 "   "

   The programmer facilitates on-board programming for the memory modules 3061 and 3086 respectively. Personality cards will later on be offered for programming other types of memory circuits used in DataBoard 4680. PROM-programming Single Board Computers are, until that, done by utilizing the 8071 capability for selective programming on chip-level.

The PROM PROGRAMMER-product No 8071 - incorporates:

a. Card module
b. Program driver
and option for:
c. Personality card

The PROM PROGRAMMER is run in a diskette based development system with the programdriver and user program modules on diskette files. It is simply plugged into a spare slot on the I/O-side of the backplane of your development system. The programdriver facilitates then for the optional steps in the whole programming procedure by conversational interaction with the operator.

Memory cct:s need not to be lifted up from the module. The programming is performed on the target PROM-module which is plugged into the back end of the PROM-PROGRAMMER. Yet the programming is done selectively at chip-level.

An ULTRA-VIOLET ERASE LIGHT unit is provided. It is especially designed to adapt to the module-level operation as the erasing is done module-level as well. You have no need to dismount the memory cct.s for reprogramming purposes.

SPECIFICATION

POWER SUPPLY   +5V ± 5%
                     - No external extra power. The power supply is
                       catered for through the 4680-bus. On-Board DC/DC-
                       converter is utilized for special voltages required.

SIZE

CONNECTOR   B 64 two-row connector
                     (DIN 41612) on both sides.

BUS CONNECTION   I/O-side of the 4680-bus.

PIN NUMBERING SIGNALS

The normal I/O-signalling is used. Refer to the System Manual.

Operator Manual

Each 8071 is shipped with an operator manual referred to as:

PROM PROGRAMMING, Operator Manual.

Systems requirements

A floppy based 4680-system e.g.
  a) Alfa
  b) Beta
  c) Gamma
or equivalent if the system is configured by the user himself.

Note: DataBoard 4680 facilitates easily on site programming
provided that the application system is based on the Z80 Double
Board computer.

The floppy system requires: DOS (+Bootstrap) interface + DMA
(8") or controller (5 1/4") which are included if not part of
the application.

Card selection

Code plug location:

Address code: 10Q (08H)

Bus connection

I/O-side of the 4680-bus.

Pin numbering and signals

The signal interface between the PROM-programmer and 3061/3086 is
routed on the unused side of the connector, i.e. the B-pins. The
pin assignment is as follows:

SEL	bits 0-4	pins 28B - 24B
	5-7	pins 20B - 18B
MPX		pin 17B
PROG		pin 5B
DATA	bits 0-7	pins 16B - 9B
Vpp		pin 31B

Special signalling is used for driving the personality cards.

The PROM ADDRESS

The Base Address of the PROM-module must correspond with the
address allocation of the concerned program module.

On-board programming allows use of the PROM-number as reference
for addressing the PROM's.

The maximal PROM-number available is 7, range 0 - 7.

You can also use explicit addressing with no concern to the
address allocation included in your program file.

The commands DUMP, TEST, VERIFY, and PROGRAM can be used with
either explicit addressing or by using the PROM-number. PROM-
numbering facilitates better possibilities for duplication of
PROM's when on-board programming.

Example:

PROM-number	Address range
0	4000 - 47FF
1	4800 - 4FFF
etc	

Where programming 2716 is performed on base address 4000H.

Addressing in the PROM Module

Three switches in the PROM-module selects the module base-address.
These switches correspond with the address bits A13, A14 and A15.

PROM: 2758

A15	A14	A13	Base-address
0	0	0	0000H, 000:000Q
0	0	1	2000H, 40:000Q
0	1	0	4000H, 100:000Q
0	1	1	6000H, 140:000Q
1	0	0	8000H, 200:000Q
1	0	1	A000H, 240:000Q
1	1	0	C000H, 300:000Q
1	1	1	E000H, 340:000Q

PROM: 2716

A15	A14	A13	Base-address
0	0	1	0000H, 000:000Q
0	1	1	4000H, 100:000Q
1	0	1	8000H, 200:000Q
1	1	1	C000H, 300:000Q

PROM: 2732

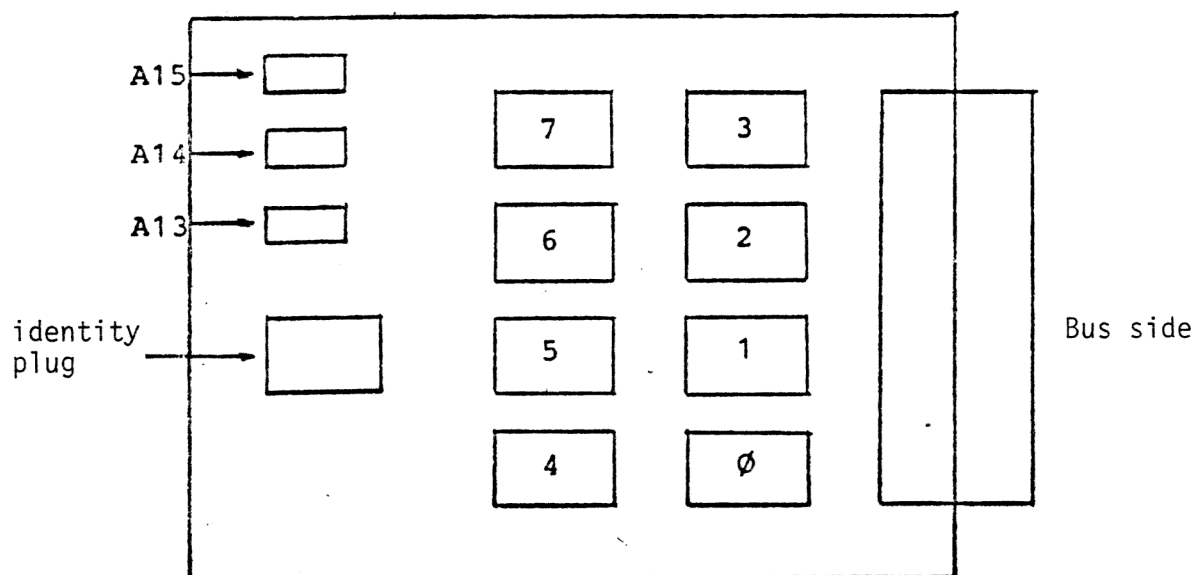| A15	A14	A13	Base-address
0	1	1	0000H, 000:000Q
1	1	1	8000H, 200:000Q

PROM-module 3061

Fig. 1 PROM-locations

Refer to data-sheets and logic schematics when defining the
identity of the memory circuit used.

Command format

Commanding uses a general format:

<command-id> <space> <parameter>...

<command-id> is a mnemonic code for a specific function called
            upon

<space> separates the command-id from the parameters if any.

... parameters are required where specified and are
            separated by delimiter (comma-character).

The command id-code does not need to be entered in the whole
extent. The programdriver analyzes only the first character
which are needed to extinguish the commands from each other.

Notes on commands

Commanding 8071 is done in dialogue where the system in all cases
gives aternatives and then determines the next sequence depending
on the significance of the answer.

In the following special characteristics on some of the commands
are clarified.

DUMP

DUMP <PROM-number>

The command is equivalent to the GET-command. In the case of DUMP
the data is transferred from the specified PROM to memory. The
command facilitates copying PROM's.

TYPE

TYPE (PROM TYPE)

PROM-type is given to call the one program required to run the
programming of the specific type of memory cct.

Enter the TYPE-command excl. parameter and the program drivers tells
you what memory-cct:s the active programdriver is able to program.
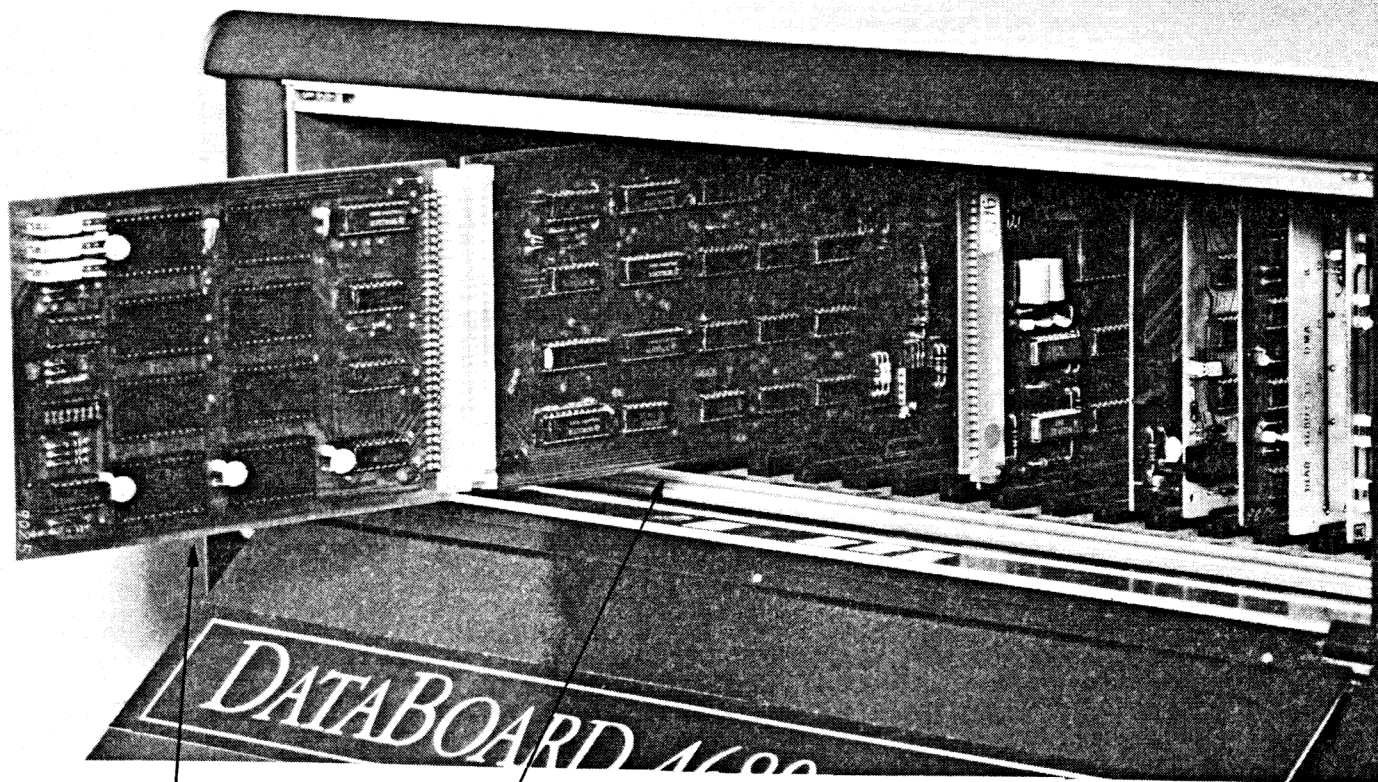
3061

8071

Plugged on the
I/O-side

The programming procedure

1. PROM-programming is performed only in diskette-based systems.
   The PROM-program-modules are both in direct access on disk-
   files and are handled as such by their file names.

2. The PROM-programmer is plugged into a free slot on the I/O-side.

3. The memory circuit used must be identified on the memory module
   itself. Check with the enclosed diagram.

4. Select the module on-board base address, which must correspond
   to the allocation of the program module to be programmed.

5. Start the PROM-programming dirver by normal DOS-commanding on
   the corresponding task-file. Standard <PROM>. Enter: PROM.

6. Check the module base address which is read and output to you
   for safety purposes.

7. Specify the data-file, your program module, by using the command.
   FIL   <File name>

   Check the address-range which is output for your check.

8. Fetch data for porgramming use GET-command.

   GET   <number>   Ex. GET 1.
         <address-range>

   <number> 0-7, 0 is defaulted if not keyed.
             The module base address is used as reference.

   <address-range> the absolute location given in hex. Ex. GET 4800.

   The system receipts by outputting the running address range of
   the running PROM.

9. Test the PROM-circuit with command TEST to check that it's erased.

10. Start the real programming procedure by command:

    PROGRAM

11. The system checks automatically that data is PROMed correctly. If
    not you'll get a fault message.

12. Data for next circuit is fetched by simply commanding:

    NEXT

    Which increments the PROM-number by 1 and loads concerned data
    from the file to the programmer-buffer.

13. Start programming as shown in point 10 etc.

14. You are able to check the programming by explicitly commanding:

    VERIFY

    which compares the buffer content with the PROM. Max 20 mismatches are output.

    If so, then switch the concerned PROM or if serious to another type.

15. When finished the whole session, key in:

    EXIT

    and you return back to DOS.