

5071

FEB 82 1 2 \*

## CONTENTS

1. Description
1. Applications
2. Technical data
3. Reference to 5071-H
4. Menu

## INNEHÅLL

1. Beskrivning
1. Användningsområde
2. Tekniska data
3. Referens till 5071-H
4. Meny

## DESCRIPTION

- The 5071 EPROM programmer board is used to transfer executable program code from a task file on diskette or from old PROM:s to EPROM placed on the 3061 or 3086 memory board.
- The entire memory board, with max 32K Bytes can be programmed directly. Alternatively selective programming on chip level CAN be used.
- The EPROM types 2758, 2716 and 2732 can be programmed on the 3061 board and the EPROM type 2532 on the 3086 board.
- For floppy-based systems only. DataBoard 4680 or ABC80/ABC800.
- The PROM-programmer software is delivered with the DOS6 and OS8 operating systems. For ABC80 or ABC800, the 5071.A version shall be ordered, including software.
- A command menu is displayed as a help for the operator. Facilities are available for testing that the EPROM is erased, for programming from diskette or a master EPROM and for verifying the EPROM. With the Command Substitution facility (CSS) the entire memory board (max 32K) can be programmed without operator intervention.

## BESKRIVNING

- 5071 EPROM programmeringskort används för att överföra exekverbar programkod från maskinkods filer på diskett eller från gamla PROM till EPROM som sitter på minneskortet 3061 eller 3086.
- Hela minneskortet, med max 32K bytes kan programmeras direkt. Alternativt kan varje EPROM på kortet programmeras selektivt.
- EPROM 2758, 2716 och 2732 kan programmeras på 3061 kortet och EPROM 2532 på 3086-kortet.
- För floppybaserade DataBoard 4680 eller ABC80/800 system.
- Programmet för 5071 levereras med DOS6 och OS8 operativsystem för DataBoard. För ABC80/ABC800 skall 5071.A versionen beställas, till vilken programmet medföljer.
- En kommandomeny visas som hjälp för operatören. Kommandon finns för att testa att EPROMet är raderat, programmera från diskett eller ett master EPROM samt verifiera EPROMet. Om kommando-substitutionsprogrammet (CSS) används kan hela minneskortet (max 32K) programmeras utan operatörens ingripande.

## APPLICATIONS

## ANVÄNDNINGSSOMRÅDEN

-Development sytem for micro-computers.

-Utvecklingssystem för mikro-datorer.

## TECHNICAL DATA

## TEKNISKA DATA

Power Supply incl. PROM  
Spänningsmatning med PROM

+ 5V +/- 5% 330mA  
(5071 contains an internal DC/DC converter for the programming voltage)

Bus connection  
Anslutning till bussen

I/O-side. The I/O card address shall be 10 octal to adapt to software.

I/O sidan. I/O-kortadressen ska vara 10 oktalt för att passa programmet.

Connector  
Kontaktidon

B 64 pin Standard Europe connector (DIN 41612) on both sides.

EPROM

2758, 2716 or 2732 on the 3061.  
2532 on the 3086 memory card.

## OPERATING MANUAL

## HANDHAVANDE MANUAL

-See the 5071-H Operating manual for installation and operating.

-Se 5071-H Handhavande-manual för installation och handhavande.

## DISPLAYED MENU AT START-UP

## VISAD MENY VID START

\*\* 4680 PROM PROGRAMMER V 2.10 \*\*

## AVAILABLE COMMANDS ARE:

FILE <FILENAME>	SPECIFY INPUT FILE
GET <PROMNR>	FILL DATA BUFFER FROM FILE
NEXT	STEP TO NEXT PROM-NR
INV	TRUE/FALSE DATA
EXPLAIN	EXPLAIN COMMANDS(THIS TEXT)
RADIX 8/16	SELECT BASE
END	EXIT TO DOS
DUMP <PROMNR>	DUMP FROM PROM TO BUFFER IN MEMORY
PROGRAM<PROMNR>	PROGRAM PROM
TEST <PROMNR>	TEST PROM (ERASED OR NOT)
VERIFY <PROMNR>	VERIFY PROM CONTENTS VS. BUFFER
TYPE (PROMTYPE)	(DEFAULT I27XX)

This datasheet information is subject to change without notice.

5071-II

FEB 80	1	7	X
--------	---	---	---

## 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

<u>A15</u>	<u>A14</u>	<u>A13</u>	<u>Base-address</u>
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

<u>A15</u>	<u>A14</u>	<u>A13</u>	<u>Base-address</u>
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

<u>A15</u>	<u>A14</u>	<u>A13</u>	<u>Base-address</u>
0	1	1	0000H, 000:000Q
1	1	1	8000H, 200:000Q

PROM-module 3061

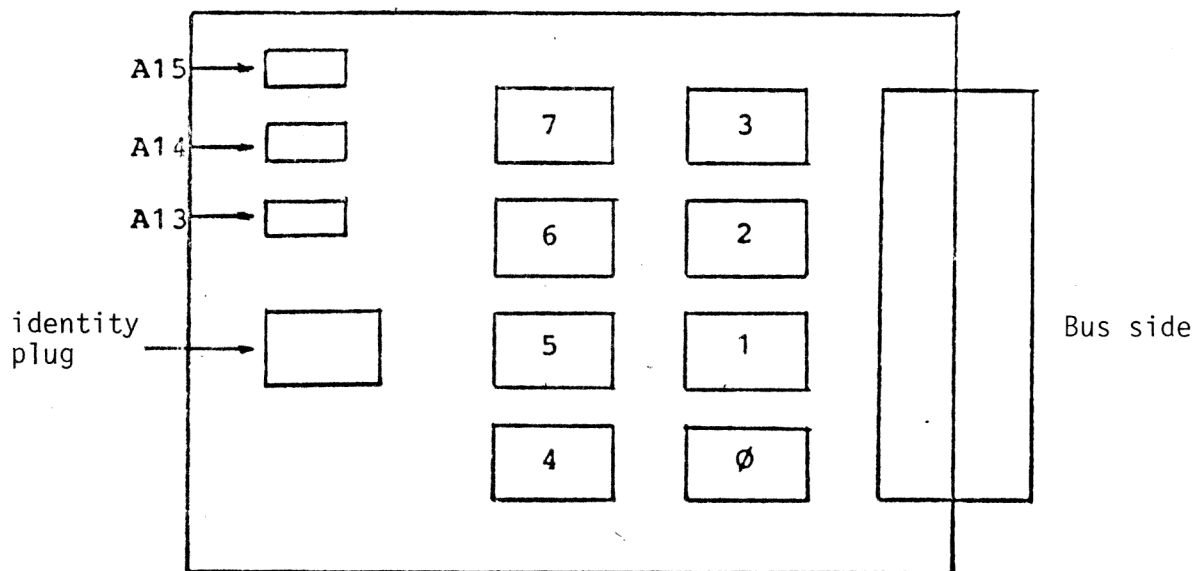


Fig. 1 PROM-locations

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.

<parameter>... 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 5071 is done in dialogue where the system in all cases gives alternatives 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

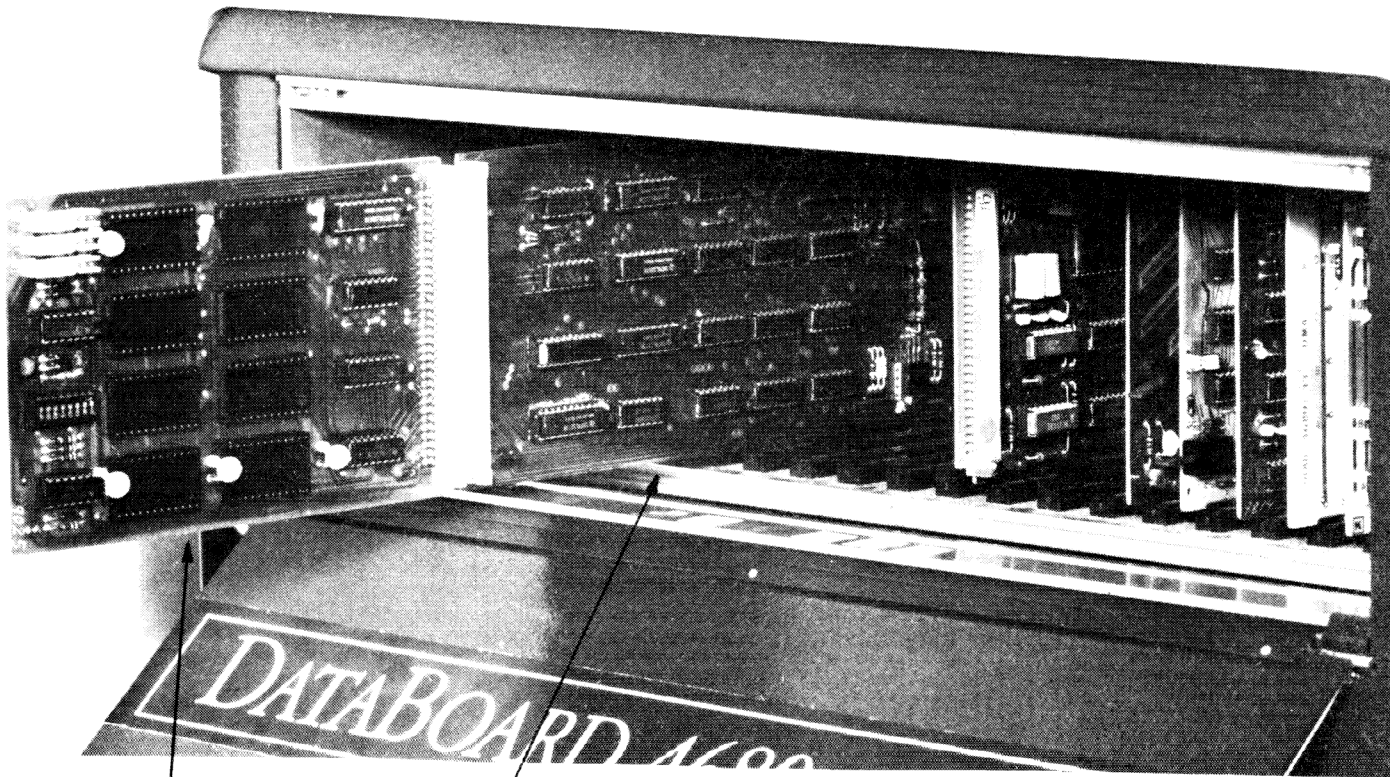
TYPE (PROM TYPE)

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.

### COMMAND SUBSTITUTION FILE USAGE

The DOS CSS-facility, having the commands on a command substitution file, can be used. This enables programming all EPROM:s on the 3061 card without operator intervention.



3061

5071

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 driver 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 programming 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.





1

## 1205

