

DataBoard 4680[®]

System Manual



SATTCO

DATA OCH KOMPONENTER.

Sattco AB Dalvägen 10, S-17136 Solna Tel. 08-734 0040 Telex S-11588

D A T A B O A R D 4 6 8 0

SYSTEM MANUAL

April 1, 1979

CONTENTS

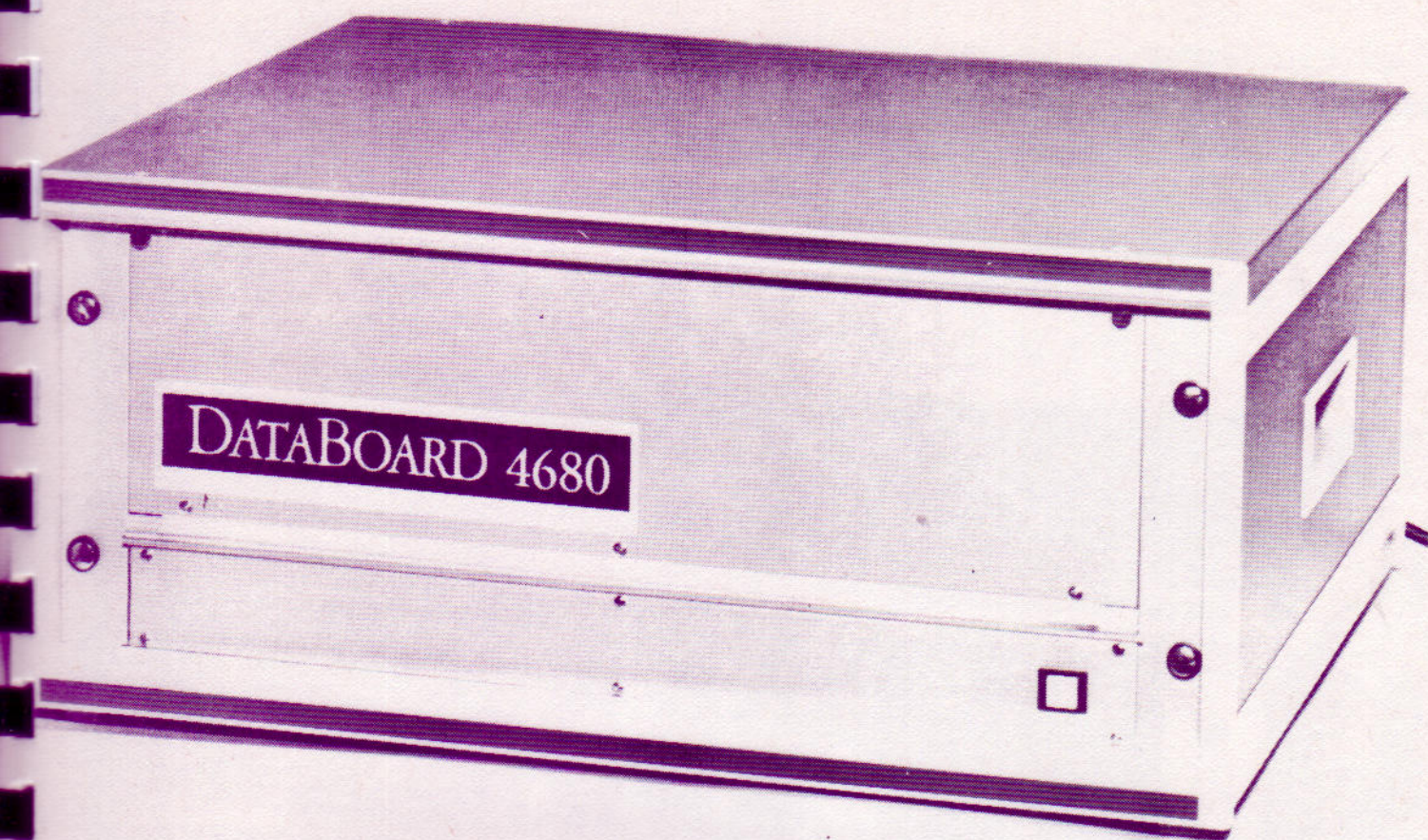
	<u>page</u>
1. DESIGN PHILOSOPHY FOR DATA BOARD 4680	1
2. INTRODUCTION TO THE 4680-BUS	4
2.1 General	4
2.2 Memory-side	5
2.3 Input/output-side	z5
3. SINGLE OR DOUBLE BOARD COMPUTER	6
4. THE 4680-BUS	10
4.1 Interface circuits	10
4.2 Signalling on the memory-bus	11
4.2.1 General	11
4.2.2 Signal descriptions	12
4.2.3 Access times	14
4.2.4 Wait-state signalling	14
4.3 Signalling on the I/O-bus	15
4.3.1 General	15
4.3.2 Function	16
4.3.3 In- and outports	17
4.3.4 Signal description	18
4.3.5 Synchronization with CPU	20
4.3.6 Code for channel selection	21
4.3.7 Examples of in/out-signalling	21
4.3.8 Bus expansion	23
4.3.9 ABC-80 expansion	23
5. THE INTERRUPT HANDLING	24
6. I/O-TRACE	26
7. RESET OF CPU AND RESTART	27
8. BUILD YOUR OWN INTERFACE	29
9. SYSTEM DEVELOPMENT	31
9.1 General	31
9.2 PROM-programmer	32
9.3 Programming	33
9.4 Development system	36
9.5 Peripherals	37

	<u>page</u>
10. SOFTWARE	39
10.1 EDITOR	39
10.2 ASSEMBLER	39
10.3 High-level languages	40
10.3.1 BASIC	40
10.3.2 FORTRAN	40
11.3.3 PASCAL	40
10.4 LINKER	41
10.5 DEBUGGER	41
10.6 DOS (DISC Operating System)	41
10.7 DOS.8MT (DOS with Multi-Task capabilities)	42
11. I/O-INTERFACING	43
12. CONFIGURING MEMORY	45
13. POWER REQUIREMENTS	47
14. POWER SUPPLIES	48
15. ASSEMBLING THE SYSTEM HARDWARE	49
15.1 Backplane	49
15.2 Power	50
15.3 Interrupt signal wiring	51
15.4 Card modules	51
15.5 Configuring memory modules	52
15.6 Configuring I/O-modules	53
15.7 Short about spare holding	54
16. A GUIDE FOR CONFIGURING A SYSTEM	55
17. TEST FACILITIES	59
17.1 Control Panel	59
17.2 The I/O-tester	60
17.3 Testprograms	60
17.4 In-built test facilities	61

	<u>page</u>
18. TRAINING	62
18.1 Off-shore training	62
18.2 On-site training	62
19. ORDERING FROM DATABOARD	63

Pictures

Pic. 1	DataBoard 4680 system mounted in a desk box	1
Pic. 2	System built-up in a 19" rack	2
Pic. 3	The single board computer 1057	6
Pic. 4	The double board computer 1043/1044	7
Pic. 5a	14-slot rack	9
Pic. 5b	Back plane of 14-slot rack	9
Pic. 6	Memory module 2056 - 16 K byte RAM	11
Pic. 7	I/O-module 4009-8 Transistor Power Outputs	16
Pic. 8	PROM-programmer	33
Pic. 9a	Floppy disk based system	34
Pic. 9b	Minifloppy based system	34
Pic. 10	DataBoard 4680 control panel	59
Pic. 11	DataBoard 4680 I/O-tester	60



Pic. 1 DataBoard 4680 system mounted in a desk box.

1. DESIGN PHILOSOPHY FOR DATA BOARD 4680

The fast evolution of the LSI technology and microprocessor technique leads to difficulties for small and medium-sized organisations to economically and swiftly offer the latest possible technical performance. The main competitive factor in the future will therefore be the ability to rapidly use new techniques. Low cost and the need to avoid complete reconstruction are of great value.

Dataindustrier AB has met these problems from the very beginning when microprocessors became available in the early 1970s. Hence the goal was set to make a logically modular design which would make it easy to adopt new microprocessors (CPU:s), memory circuits etc. without reconstruction. A logically modular design gives a system well suited for maintenance and troubleshooting.

The result is a microprocessor-system built up on a general bus - the 4680-bus - divided into an I/O-side and a memory-side. On each side the cards can be mixed in any combination. See figures 1 and 3.

Every card module is built up on Euro-card standard size, 100x160 mm. The system is put together on modular basis. It is configured in standard racks on the bus in a backplane.



Pic.2 System built-up in a 19" rack.

The number of available modules is extensive. By selecting the components you need, you can put together a system - small or big - ideally suited to your particular application and packaging requirement. New modules are being continuously developed.

Several different microprocessor CPUs can be used, namely:
Intel 8080, Motorola 6802, Synertek 6502, Zilog Z80 and Signetic 2650.

The modularized build-up has given a system which is simpler for test and service than would have been the case if a design had been chosen

with all components on the same board. Great flexibility on system size has been gained. A small system is not burdened by a heavy initial cost. Subsequent developments can be introduced without the need to change the whole system structure. The cost of succeeding developments is low and therefore keep DataBoard 4680 pacing the evolution of technology. 16-bit microprocessors will be implemented as soon as they become available contributing more data power for DataBoard 4680.

2. INTRODUCTION TO THE 4680-BUS

2.1 General

The bus structure provides a common element for communications between the different functions in a computer system. The bus gives a standard interface which is necessary in a highly modularized system such as DataBoard 4680. A wide variety of system modules are provided. Knowledge of the 4680-bus is essential in order to understand the system structure and to understand the design of the individual modules. The user must have a basic understanding of the 4680-bus to configure a system. A detailed description is therefore made in chapter 4. Chapter 9 deals with the electrical interface requirements for users building their own interfaces.

The 4680-bus is designed to use different types of microprocessors. It is a parallel, bi-directional and asynchronous non-multiplexed tristate bus. A data rate of 2.0 MHz is possible.

The bus is divided into two separate sections, a memory-side and an I/O-side. This gives great flexibility in choosing CPUs when in several cases, different methods of memory-addressing and different I/O-processing are applied. The CPU board is placed in-between the two sections. Both sections include control lines which direct and control the dataflow and the processing functions.

The CPU can be one of two types, namely a single board computer or a double board computer. Figures 3 and 4 illustrate configurations on these. Address lines run to all units directly addressed by the CPU, that is, the memory modules and the control unit. All units are connected to the data lines. Each section is controlled by separate groups of control lines.

The double board computer configuration contains a special Control Unit for controlling the I/O-side of the bus. Special lines are drawn for each one of the commands which control the I/O. The commands - which the CPU orders by selecting in or outports - control the interface units and also the system functions in the Control Unit.

On the memory-side the CPU accesses different types of memories such as ROM, (E)PROM, static RAM and dynamic RAM. DMA is also located on the memory-side. The control lines comprise required signals for control and synchronization of the dataflow. The DMA has special lines to work independently from the CPU. It has direct communication with its I/O-interface unit.

The single board computer is designed for small applications. The system is based on the internal on-board memory and I/O-interfacing. Expansion is done in accordance with the double board computer through the 4680-bus. The single board CPU contains all bus interfacing, but less interrupt levels and only static memory may be used. This bus interface forms the basic part of the 4680-bus and details can be studied for all bus signals in figure 2.

Figure 5 illustrates examples of the functional flow from the bus into the card for both sections.

2.2 Memory-side

Includes 16 address lines and 9 control lines, 64 kbytes of memory can be configured through standard modules and connected to this side of the bus. Each card has its own logical address code (switch or jumper selectable) for flexible allocation of the memory module within the 64 kbyte memory-area. There are thus no restrictions on bus-locations. The memory can be any combination of RAM, ROM and (E)PROM.

2.3 Input/output-side

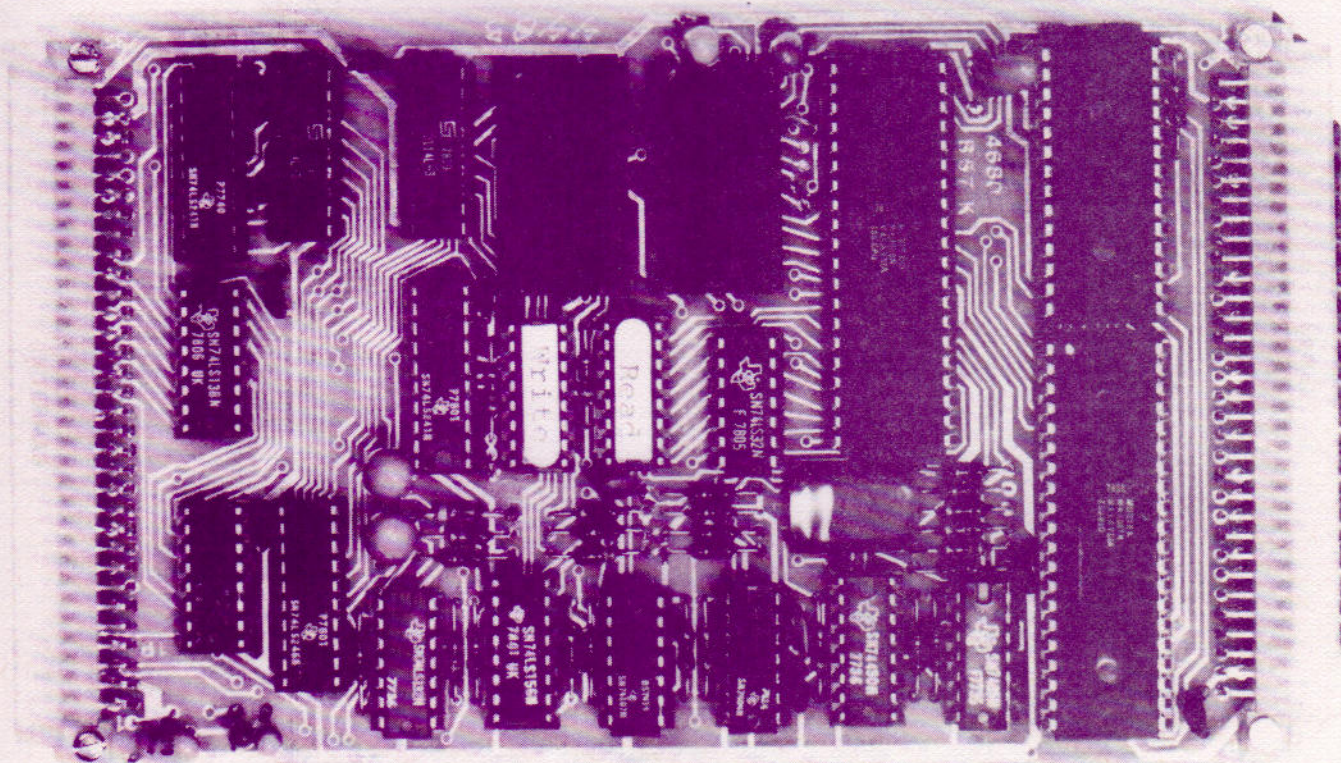
Contains 8 data lines and 10 control lines. The maximum number of interface units which can be connected is 64. Each unit has its own logical address code, jumper-selected on the bus. The card position is determined by choice. The input/output-side is designed to be run by commands by which the desired input/output functions inside the interface cards are selected and executed. A simple and standardized synchronisation between CPU and I/O-modules is thus provided.

3. SINGLE OR DOUBLE BOARD COMPUTER

The choice between these two computers is dictated by the system and functional demands of the particular application. It is a question of how rigidly the system is defined. If the system development is made on the application system, the double board computer is chosen. If the single board computer is chosen you still have flexibility by configuring the system in the backplane of the development system. A jumper-board is used to interface the CPU to the memory-side. See figure 7.

The two computers are compared with each other on the basis of the internal on-board functions as follows:

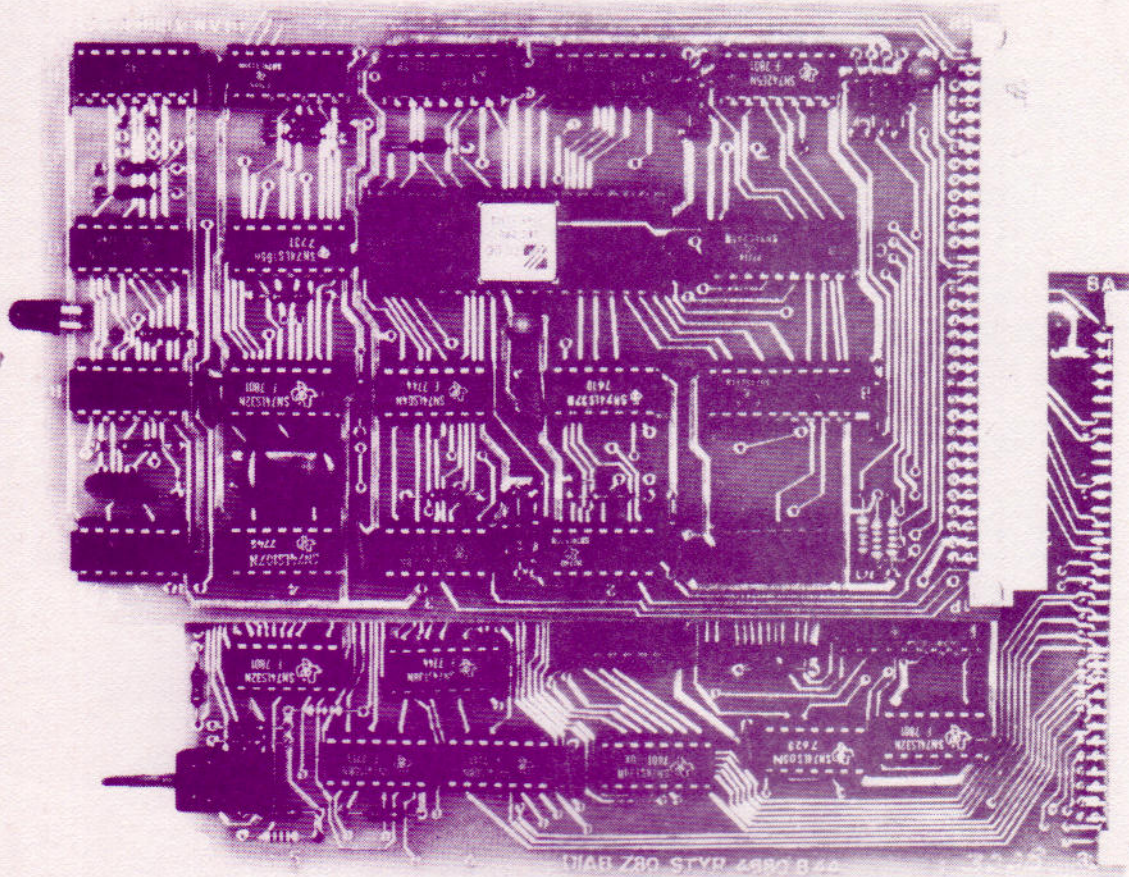
The single board computer



Pic. 3 The single board computer 1057.

Allows applications that only use the on-board memory- and I/O-interfacing. System hardware can be configured with a minimum number of modules.

The double board computer



Pic. 4 The double board computer 1043/1044.

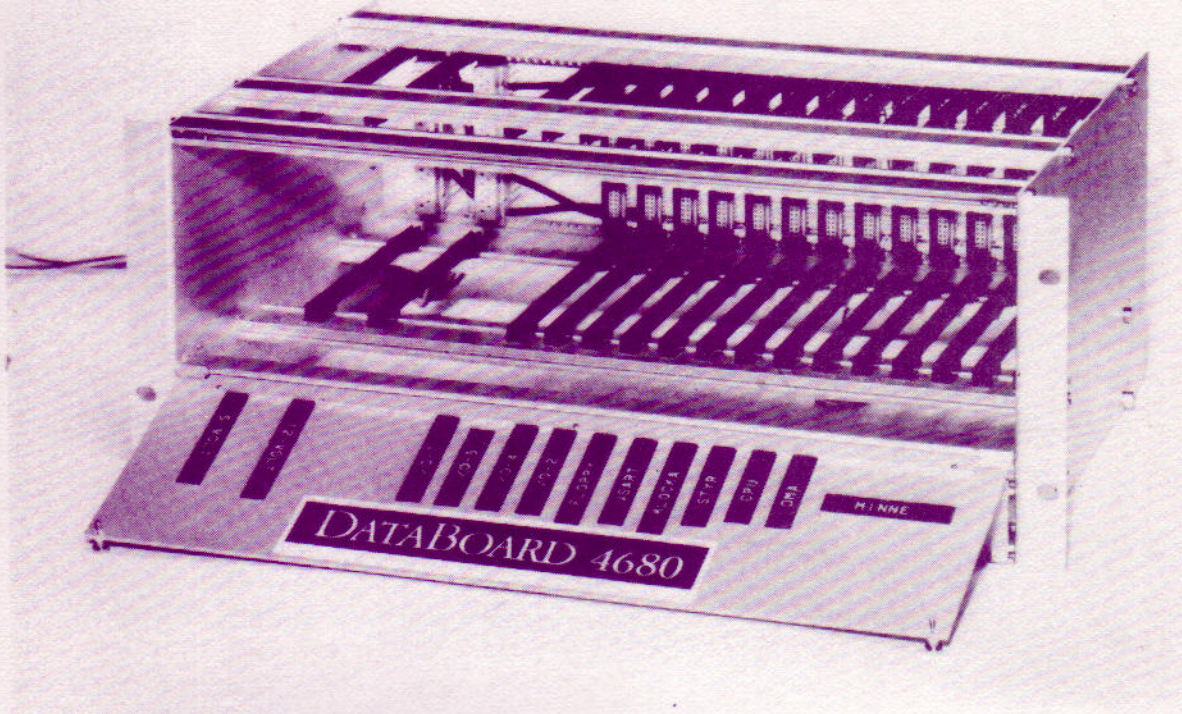
Supports DMA and dynamic RAMs. It is able to handle slow CMOS memories. Contains increased capacity for I/O interrupt handling. Provides programmed restart which is separate from the main memory. Several bootstrap programs called "mini-boots" are available as options.

In favour of the single board computer

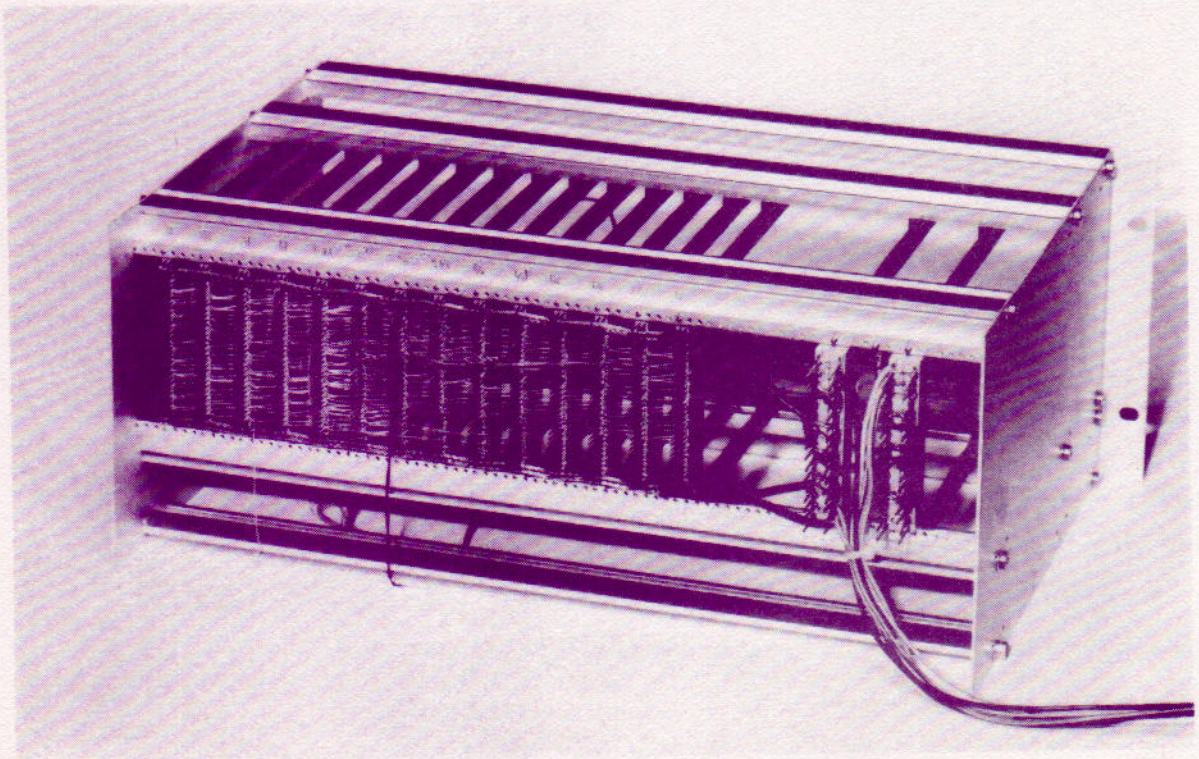
- Memory requirements are met by the on-board memory (RAM and PROM).
- I/O interfacing is done by on-board I/O.
- The I/O signalling is handled without interrupts or where no more than two interrupt levels are needed.

In favour of the double board computer

- Interrupt handling on several priority levels.
- I/O-trace. (Saves the last out-operation).
- Programmed restart is separate from the main memory.
- Slow memory and/or slow I/O may be used.
- Supports both dynamic and static memory.
- A large memory may be accessed.
- DMA.
- Accommodates the complete range of I/O- and memory-modules.



Pic. 5a 14-slot rack.



Pic. 5b Back plane of 14-slot rack.

4. THE 4680-BUS

The bus is divided into two sides, namely the memory-bus and the I/O-bus. All inherent signals are defined. Physically the signals form the backplane design as shown in picture 5 for the double board computer. Figure 6 presents the inherent threads and signals. Figure 7 shows the jumpers required to attach the single board computer to the same backplane.

4.1 Interface circuits

There are three basic elements in the bus interface, namely:

- address decoders
- bus drivers
- control signal logic

Address decoders

Every card connected to the 4680-bus has to be identified. With respect to a memory card it must be selected in allocated memory-area when addressed. This is done by switches or code plugs which select the beginning of the address range.

An I/O-card is selected differently but also uses code plugs for the unique card-address.

Figure 20 shows and describes the use of code plugs and switches.

Bus drivers

The 4680-bus requires tristate drivers to the bi-directional data lines. Destinated logic that solely receive data interface to the bus with buffers only. Tristate drivers are required when data are placed onto the bus. When both the read and write functions are required then parallel bi-directional bus drivers are used.

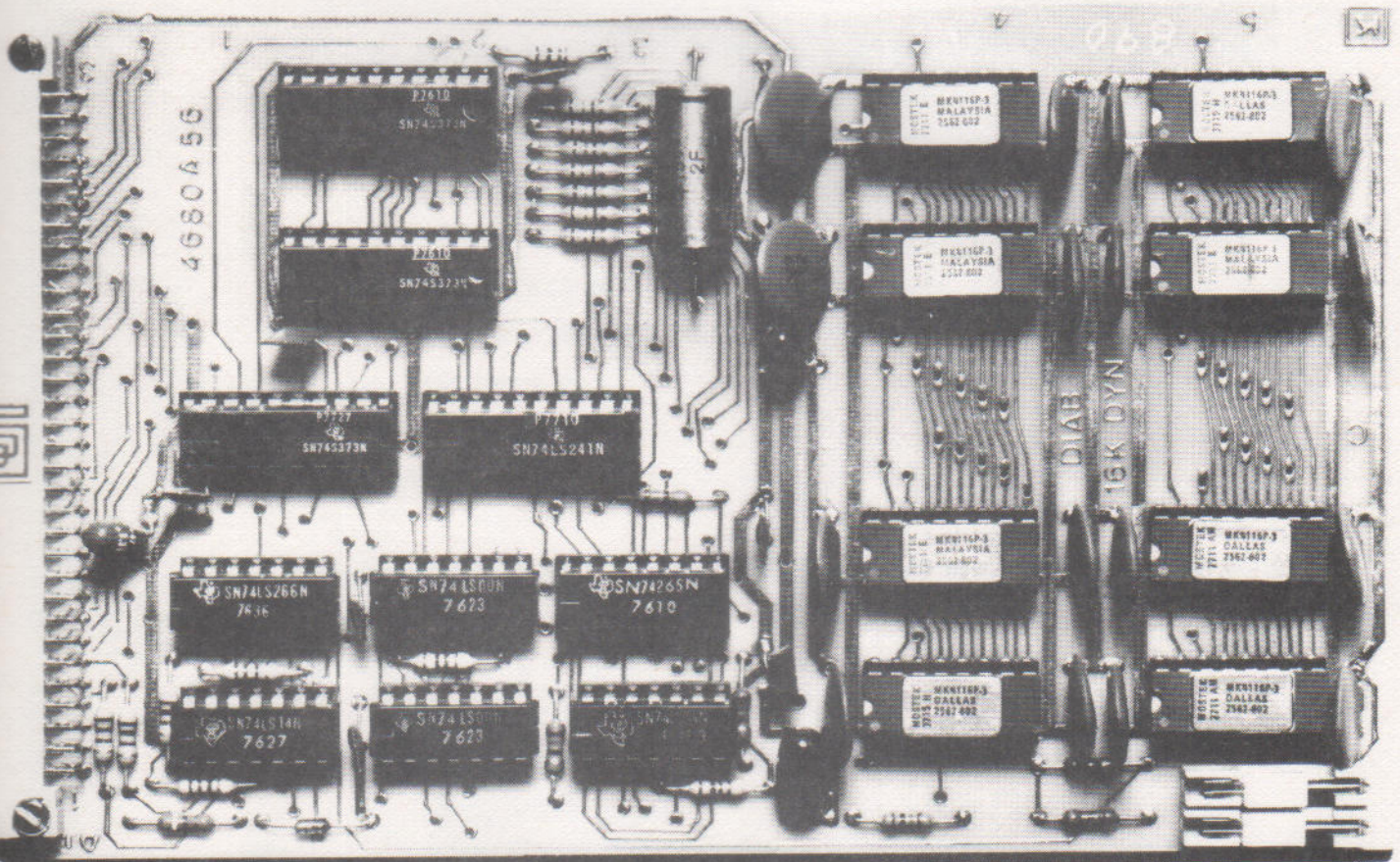
Control signal logic

Consists of the circuits that control card functions and direct the dataflow. Most of the control signals are active-low. This is denoted by an asterisk on respective signals in figure 6. The low level of the control signal indicates active control.

4.2 Signalling on the memory-bus

4.2.1 General

All memory modules are available for use with the Z80-double board computer. The single board computers use only static memory, namely static RAM, ROM and (E)PROM. Dynamic memory is only handled by the double board computers.



Pic. 6 Memory module 2056 - 16 K byte RAM.

4.2.2 Signal descriptions

Associated bussignals for control are called by common names as follows:

"MEM control" for memory-access in general

"DYN MEM control" for controlling dynamic RAM

"DMA control" for controlling "direct memory access".

"MEM control"

Contains following lines:

W* - from CPU, commands write

MEMFL* - from CPU, commands read

MEMRDY - to CPU, request for wait-state(s) by MEMRDY*

Static memories only require the above control signals.

Signalling on the memory bus comprises also additional lines. Some of them are only drawn to the DMA-slot in the backplane. They are:

MI - states the instruction fetch cycle.

MREQ* - states that the address-bus contains adequate address for memory access.

READY - signals request for unlimited wait from CPU.

Ø - system clock.

"DYN MEM control"

Contains the following special control signals from CPU cycle generator:

CSTOP* - signals stop of the memory cycle.

RASST* - starts address strobe (RAS).

ADMX* - multiplexes incoming address of column selection and starts the strobe (CAS).

REFR* - states that the running memory cycle is a refresh.

The access requires furthermore the basic control signals as stated for "MEM control".

"DMA control"

The DMA of the 4680-system is the same as that used in mini-computers. DMA is not available on single board computers.

Static memories are accessed directly with their own cycle generation. The memory cycle is firmly resolved. The data sheet on the DMA-processor states the maximum access time of the memory modules. The CPU cycle generator is used when dynamic RAMs are accessed. The user determines the total access time by adding the access time of the memory circuit. The MEMRDY-signal has no relevance to DMA. The system delays include bus transfer plus drives (3*TL).

The following lines are used for communication between the CPU and the DMA-processor: memory modules are designed to work with the following access requirements, which the user will consider when selecting memory:

- HOLD* - requests cycle stealing
- HOLD A* - CPU acknowledges request
- FLOAT (DB FLOAT)* - DMA inhibits the data drivers in CPU
- (ADDR FLOAT)* - DMA inhibits the address drivers in CPU.
- XCYCL* - starts the CPU cycle generation for the access of dynamic memory.
- Ø* user - system clock

The DMA-processor is in direct communication with the controller and the I/O-device. The signal lines are built into the backplane and consist of:

- R/W* - indicates the direction of dataflow
- TREN* - transfer enable to I/O
- TRRQ* - transfer request from I/O
- PRAC* - processor acknowledge to I/O
- PREN* - DMA enable from I/O

The DMA-processor can be connected to several controllers which request access and are granted DMA one after the other. The user is not concerned with this matter when making a choice from the normal 4680 range of memory. The DMA-processor can be connected to several controllers which request access and are granted DMA one after the other. The user is not concerned with this matter when making a choice from the normal 4680 range of memory. The DMA-processor can be connected to several controllers which request access and are granted DMA one after the other. The user is not concerned with this matter when making a choice from the normal 4680 range of memory.

4.2.3 Access times

The total access time is dependent on the particular CPU used, the type of memory circuit and system delays.

Each memory module is based on its particular memory circuit. Compatible types are still allowed. The user may therefore choose different types of memory circuits for a particular memory module. The data sheets on the memory modules do not specify any access time. The user determines the total access time by adding the access time of the memory circuit chosen to a standard 100 nsec. for system delays. The system delays include bus transfer plus drives ($3 \times \text{TTL}$).

In general the memory modules are designed to work with the following access requirements, which the user will consider when selecting memory:

- supports DMA
- no DMA, fast memory circuits and no wait-state signalling needed.
- no DMA, slow memory circuits and wait-state signalling required.

The DMA data sheets specify the access time required for DMA-transfers. The user will be concerned with the 3rd case when dealing with the "early" types of memory modules. See section 15.5 Configuring memory modules.

4.2.4 Wait-state signalling

The CPU demands the memory access to be accomplished within the time related to the processor time states. The access cycle of the memory module and system delays has to match the CPU memory cycle. Where this is not the case then there has to be means of setting CPU in wait or to avoid the use of slow memories. The user is not concerned with this matter when making a choice from the normal 4680 range of memory modules. The wait-state signalling is an obsolescent function from the early days of DATABOARD 4680 when available memory circuits were slow and differed in access times. Today the situation has improved. This fact is evident on the single board computer which has no wait logic for memory. The double board computer on the other hand must provide this facility, as it covers a broader range of applications and is designed to work with all types of 4680-modules.

All standard DATABOARD 4680 memory modules provide optional wait-state signalling. The wait-state request is done with the signal MEMRDY* which is connected to the bus with on-board jumper. The MEMRDY*-signal is initiated by the card-selection logic and is active as long as the address stands high. The wait-logic on the CPU-card determines the number of wait-states.

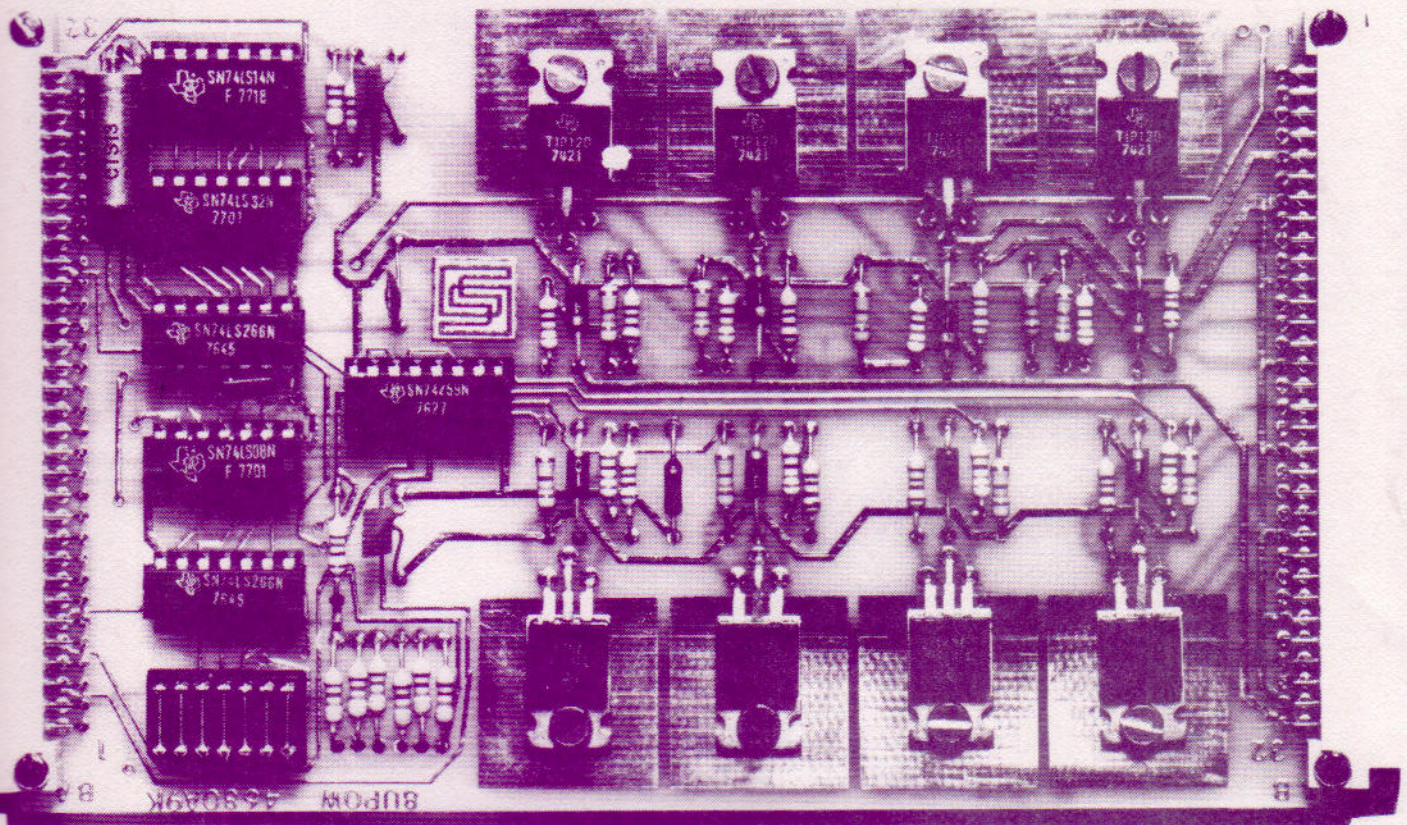
A maximum of two wait-states are available. The dynamic memory modules are normally designed to request one wait-state. The Z80 double board computer provides alternatives by on-board jumper for:

- a) Wait-states at instruction fetch only
- b) Wait-states at all memory requests

4.3 Signalling on the I/O-bus

4.3.1 General

The in/out-signals for control are referred to as "I/O-control" in the data sheets for the CPUs. In addition to these control signals the I/O-bus consists of lines for interrupts and internal standard and user-optional signalling between the modules. See the right-most side of figure 6. The bus interface on the card is shown in figure 8.



Pic. 7 I/O-module 4009 - 8 Transistor Power Outputs.

4.3.2 Function

The I/O-bus is asynchronous. It contains 8 data lines and 10 control lines. A maximum of 64 cards can be connected. The connection is done with the command CS and an address which uniquely identifies the card to be selected, which will then be the only one to accept the following commands. Cards are connected to the bus one at a time. The data bits 5-0 of the A-register determine which card is to be selected. And the command CS orders the selection. The procedure is as follows:

1. The CS flip-flop is switched ON.
2. The LED on the card is turned on to indicate that it has been selected.

3. The CSB-signal is sent from the card. It signals to the bus expansion card, if the selected card is located in an expansion rack, to turn around the data bus when an input is done.
4. The following commands are enabled on this card only.

All signalling on the I/O-bus is directed to the selected card until a new CS-command with a different card address is ordered. When this occurs the card is disconnected from the I/O-bus and a new card will receive the commands. The effect is:

1. The CS flip-flop is switched off.
2. The LED is turned off.
3. The CSB-signal is removed (returned by the next selected).
4. All commands to the card are prohibited.

The related functions are shown by the block scheme in figure 8.

4.3.3 In- and outports

The programmer deals with the I/O-commands as in- and outports. A command on an inport is made up of an I/O-input operation and an assignment of the source (inport). Correspondently a command on an outport is made up of an I/O-output operation and an assignment of the destination (outport). Hence the commands are committed to certain I/O-functions. The following table shows the reserved in- and outports (the system in- and outports). The notation "channel" is used instead of "card" to refer more correctly to functions utilising several cards. See instruction formats in chapter 4.3.7.

IMPORTS

INP 0 Fetch data from selected channel, referred to as INP or INP DATA
 INP 1 Fetch status from selected channel, referred to as STAT or INP STAT

INP 5 Connect the restart program in the Control Unit.
 INP 6 Reset the Watch Dog.
 INP 7 Reset the I/O-system.
 INP 8 Read status from DMA.

OUTPUTS

OUT 0 Send data to selected channel, referred to as OUT or OUT DATA
 OUT 1 Select channel, referred to as CS
 OUT 2 I/O-command 1, referred to as C 1.
 OUT 3 I/O-command 2, referred to as C 2.
 OUT 4 I/O-command 3, referred to as C 3.
 OUT 5 I/O-command 4, referred to as C 4.
 OUT 6 Disconnect the restart program in the Control Unit.
 OUT 7 Load interrupt mask.
 OUT 8
 OUT 9 Load the DMA controller
 OUT 10
 OUT 11

The I/O-trace memory, Control Unit option, is read with the instruction "INP xxH". xx stands for 80H + output number.

4.3.4 Signal description

RST

This signal is sent at power-up or when a command on inport 6 is ordered. All interface modules (not controlled by the channel selection) are reset.

INP and STAT

These two signals are both commands on inports and when ordered read respectively data and status from the selected channel.

OUT, C 1, C 2, C 3 and C 4

These signals are commands on outports. OUT, C1, C 2 and C 3 send data to the selected channel.

9 C 4 is used to control the selective interrupt from a channel with an interrupt facility. Data bit 7 = 1 for ENABLE. Data bit 7 = 0 for DIS-ABLE. When a channel has several interrupt sources then each is controlled by the data bits 6-4. Example: "data ready", "request to send", "received line indicator" etc.

CS

This signal is used to select channel, connect one controller to and the others off the 4680-bus. The data bits 5-0 comprise the channel address. See "channel selection".

The interrupt from the controller

An interrupt is signalled externally into controllers requiring immediate service from CPU. The appropriate priority level is wired in the backplane. The priority is thereby fixed to a particular slot position and does not follow the card when it is moved to another position. Mark in mind that it is possible, naturally, to connect several controllers to the same interrupt level. Identification of the interrupt is therefore a question of system design.

The other lines to CPU

A nonmaskable interrupt, NMI, is available for connection by wiring it into the backplane. The user therefore has the opportunity to connect a certain I/O-function to the highest interrupt level and the fastest response.

RST IN, called "reset in" in the CPU data sheets, makes it possible for the user to provide for his own control of the reset and restart from a panel or from an I/O-function.

Internal bus signals

CSS

This signal is used to transfer the channel selection from a primary to a secondary card (or secondaries) in a multi card controller. Only one is furnished with the comparator and CS flip-flop. When the CS flip-flop is switched on then CS is signalled to the other card(s) and enables the commands in the whole controller.

CSB

CSB is used for expansion and is described separately in the section on "bus expansion" below. Applications for the Z80-"double board computer" use this signal to signal for wait-state(s) when slow I/O-access.

Options

The I/O-bus contains reserved lines (OP0-OP5) used by the user for the purpose of inter-signalling between his own cards. The system uses these pins in special cases such as DMA.

The user has the opportunity of using the spare bits of the data word at channel selection to designate for example a group of signals in the controller. This usage is done in the controller 4008 the commands of which are shown below.

4.3.5 Synchronization with CPU

The times for the CPU I/O-access have to fall within the CPU time states. The required access-time is specified in the CPU data sheets. The I/O requests a wait-state(s) through the signal IORDY which is drawn to the slot position "USRT" in the backplane of the "double board computer". In applications for the Z80-double board computer the controller uses the CSB-signal to initiate this request.

The required total access time for I/O-operations is specified in CPU data sheets. This includes all delays (3TTL + bus). Normally, with the standard DATABOARD I/O modules, the user need not be concerned about access times and I/O wait-state signalling.

4.3.6 Code for channel selection

Every controller identifies itself with a code plug which is coded for a unique address. This address is compared with the incoming data bits 5-0 at command CS (channel select). See figure 8.

4.3.7 Examples of in/out-signalling

Figure 9 shows a typical signal sequence.

The signals involved require the following commands converted to the programmer's code:

OUT 1	selects card A (register A = channel number.)
OUT 0	sends data to card A, register A = data.
OUT 1	selects card B (register A = channel number.)
OUT 2	orders command C 1 to card B.
OUT 0	sends data to card B, register A = data.
INP 0	fetches data from card B, register A = data.

Every standard controller is described in its data sheets with respect to the commands it uses. This signal description is used by the programmer to design the program (driver) which is required to control the respective I/O-interface and its external signalling.

Examples on these signal specifications are shown below. The channel selection is always required and is therefore understood as implied.

Instruction formats

Input commanding on "16 I OPT"

Assembler	IN n	DATA TO REG A
	INP n	DATA TO REG A
Fortran	INPUT (n)	DATA TO VAR
Basic	$\text{B} = \text{INP (n)}$	DATA TO VAR

Output commanding on "8 URE"

Assembler	OUTm	DATA FROM REG A
Fortran	OUTPUT (m)	DATA FROM VAR
Basic	OUT (m, data)	

Note 1) n stands for inport number, n=0 in this case

2) m stands for outport number, m=0 in this case

"16 I OPT"-controller 4008

(16 optical outputs)

INP DATA Fetches data, 8 bits

Group selection is done with D 7 at channel selection.

D 7 = 1 = group 1, bits 7-0.

D 7 = 0 = group 2, bits 15-8.

The other in/out-ports are not used.

"8 URE"-controller 4007 (8 relay outputs)

OUT DATA Controls the relays.

D 2 - D 0 = Binary address of the relay which is switched on or off.

D 7 = 1 = ON

D 7 = 0 = OFF

Only one relay at a time can be controlled.

OUT C 3 Switch all relays off.

The other in/out-ports are not used.

"16 U 8 I"-controller 4005 (16 digital outputs and 8 digital inputs)

INP DATA Fetches the data.

OUT DATA Sends the 8 bits of data to group 1.

OUT C 1 Sends the 8 bits of data to group 2.

The other in/out-ports are not used.

4.3.8 Bus expansion

A bus expansion card must be used to amplify the bus signals when a large number of controllers (expansion of rack(s)) are connected to the bus or when signalling at a long distance from the CPU. As the bus is bidirectional the bus must be turned at input commands (INP). The CSB-signal is used for this purpose. The example in figure 10 illustrates such a case. Cards C and D will when selected turn the bus but not the cards A and B.

4.3.9 ABC-80 expansion

ABC-80, Advanced Basic Computer for the 80:s, is a universal micro-computer that provides a 4680-bus connection. The ABC-80 capability and application range is thereby expanded considerably from the broad range of standard DATABOARD 4680 products, memory- and I/O-modules available.

The bus interface conforms to the basic 4680-bus utilized by the DATABOARD 4680 single board computers. DATABOARD 4680 provides a special box-contained rack with backplane and power supply. A special cable for connecting the expansion box to ABC-80 is available. The 6-slot backplane provides expansion for two memory modules (static RAMs or PROMs) and 3 I/O-modules.

5. THE INTERRUPT HANDLING

This is described with reference to the Z80 - "double board computer". The interrupt handling comprises 1 nonmaskable interrupt and 8 masked interrupt levels.

When the controllers send an interrupt signal they also generate an interrupt vector. The vector is dependent on the interrupt mode selected by the programmer. The vectors are explained as follows with respect to selected mode. In each case a subroutine call is executed.

Mode 0 (8080 A mode)

The vector is delivered as an RST-instruction by the Control Unit. Eight fixed locations, determined by the Z80 CPU, are pointed to.

The CPU interprets the received RST-instruction as an instruction equivalent to a single byte subroutine call. The program execution continues at a low memory address as specified below.

Mode 1

Common interrupt level with the vector 0038H is pointed.

Mode 2

The vectors, delivered by the Control Unit, point to ~~subroutines~~ in a memory sector (8 most significant bits) which the programmer has chosen by loading register I. Eight fixed locations are provided. The Z80 CPU will combine the I-register content with the interrupt acknowledge vector - delivered by the Control Unit (1044) - to form a 16-bit address ~~containing~~ interrupt address vector table. The CPU will then execute a call to the memory location pointed to by the interrupt address vector. These 16-bit addresses identify the first executable instruction of the interrupt service routines. The table contents must be created by the programmer.

The nonmaskable interrupt (NMI) calls subroutine at address 0066H.

The fixed vectors are as follows:

Mode 0

Level	0	1	2	3	4	5	6	7
Vector	00H	08H	10H	18H	20H	28H	30H	38H

Mode 2

Level	0	1	2	3	4	5	6	7
Vector	C7H	CFH	D7H	DFH	E7H	EFH	F7H	FFH

A lower priority level is interrupted by a higher level. Level 0 is the highest maskable priority level. Programmed masking is available. The mask is loaded with the command "OUT 7 [value]". The loaded value (0-7) determines the level from which, in descending order of priority, the interrupt is disabled. "OUT 7, [value = 0]" inhibits all interrupts (not NMI). "OUT 7 [value = 8]" allows all interrupts.

If the interrupt signal is too short and there is no time to read the actual level, then level 7 is automatically chosen. This the designer may avoid by using normal procedure, which is to turn off the interrupt signal with a receipting command.

Every controller which has an interrupt function is also provided with an enable/disable of the interrupt signal. The programmer activates (enables) or "kills" (disables) the interrupt function selectively on each channel. See figure 12 with respect to conventions in signalling.

6. I/O-TRACE

This is available on the double board computer and comprises 64 byte of RAM for automatic save of last operations on the outports. When a command on an outport is recognized, the databyte on the data-bus is saved simultaneously in the trace-memory at a corresponding address as the respective outport number. The trace-memory is read with the command "INP X", where X stands for [80H + " the number of the actual outport"]. The "channel select" is on outport 1 and consequently read with the command "INP 81H". Tracing ranges the outports 0 - 3FH.

The last event - operated out function - is thus traced. The trace-memory contains an image of the running I/O processing. It provides for restoring for example channel selection at levelled interrupt service. Spare outports may be used for intermediate storing. These dummy out operations result in trace only and will not affect I/O.

7. RESET OF CPU AND RESTART

The reset of CPU and subsequent restart is available through:

- a) Manual command
- b) Bus signal
- c) Power-up
- d) System monitoring through hardware (Watch Dog)

Manual command

In the case of the "double board computer" reset is ordered by activating a switch on the Control Unit.

Bus signal

Reset is generally provided by wiring into the 4680-bus. The user is therefore free to choose any technique in initiating his reset-signal. The normal and simplest way is to wire wrap a switch on a panel, directly to the Control Unit slot position of the bus. The principle is shown in figure 2.

Power-up

The reset signal is generated by a circuit containing a large electrolyt in the Control Unit of the "double board computer".

System monitoring by hardware - Watch Dog

The Watch Dog works on a time-out basis which occurs if the command "INP 6" is not programmed in at least once a second.

The Watch Dog provides a way to disconnect programs that have gone wrong or occupy too much processor time.

The function is optional and is, in the case of the double board computer, controlled by a switch on the Control Unit which disables or activates the Watch Dog-function. A "kill dog"-wiring on the bus has the same purpose in the case of the single board computer. A wire wrap may be used to inhibit the Watch Dog. See figure 2.

Function

The restart function is in all cases a "reset of CPU". The reset signal is generated to both the CPU and to the I/O interfaces. The I/O-bus is however not affected in the case of Watch Dog reset. A programmed reset is generally available by a normal CPU instruction.

The restart program

The program starts on address 0 at restart. A special restart program is available with applications of the "double board computer".

This is contained internally in a PROM-program located within the Control Unit from where it may be switched in or out of the system. If the restart program is enabled it will start automatically at reset. It can also be started with the command "INP 5" and turned off with the command "OUT 6".

The following procedure is normally utilized at system turn off. The instructions "OUT 6" to jump to the start point of the ordinary program is pushed into the stack. The stack is then addressed and the instructions are executed in the named order.

8. BUILD YOUR OWN INTERFACE

On the memory side of the bus the functional need for most applications is well covered by standard modules. The I/O-side on the other hand depends completely on the user signal interface which usually differs from application to application. In most cases the broad range of standard modules provided cover most interfacing needs but there are often some left for the user to design himself. This chapter is not meant to be comprehensive but will answer some questions arising from this type of work.

In general an I/O-card consists of three parts which are physically laid out in that order. The three parts are:

- 1) Bus driving and listening
- 2) Control of the actual function
- 3) External signalling of the function itself

The first part differs in only a few signals. The second part is tied to the functional aspects of processing. The third part differs with respect to signal levels, number of signals etc.

The DataBoard 4680 provides two prototype cards to help the user design his own I/O-interfaces. One is blank and provides complete freedom of design - from bus to external signalling. The second provides the bus interface - the first part is described above and shown in figure 8. Most bus signals except for optional ones are interfaced and ready for use. The designer may thereby avoid bus-problems and can concentrate on the main task - the function - instead.

The bus interface logic is based on "74 LS"-circuits.

The driving capacity must be minimum 16 mA/output.

And maximum 9,8 mA/input load is allowed. Usually the load is 0,4 mA/input.

The interrupt service must be designed for programmed switch off of the interrupt signal.

An "I/O-tester" is used to check the design. It provides the complete I/O-bus interface. The test is accomplished in the normal environment of the 4680-backplane or may be selfcontained.

9. SYSTEM DEVELOPMENT

9.1 General

The microprocessor user has the choice of using development systems offered by the respective microprocessor manufacturers. Program development can also be accomplished via the use of terminals connected to large computers. These alternatives (among others) for system development are naturally available for the user of DataBoard 4680.

There are, generally speaking, no obstacles to doing development work in an environment other than DataBoard 4680. The user has the option of entering the 4680-system with programs on PROMs and/or loaded to an I/O-media compatible with the 4680-system. The user should not, however, underestimate the importance of working in an environment which is the same from the beginning all the way through until the system is delivered and working.

The 4680-system was initially based on the Intel 8080 and a development system designed for it. The current development systems support primarily the use of the Zilog Z80.

DataBoard 4680 uses the same hardware for development as is used in the application itself. When program development is completed then the development system is converted to an application by simply plugging the I/O-modules into the backplane, and connecting them to the process interface. No special equipment such as ICE (in circuit emulator) is required.

The system field tests are supported by the Control Panel, which controls processor functions at hardware level in single step and free run mode, and the I/O-Tester, which tests the I/O in a real process signal environment.

If the user prefers one of other available microprocessors the system must be developed in the development systems for that particular microprocessor.

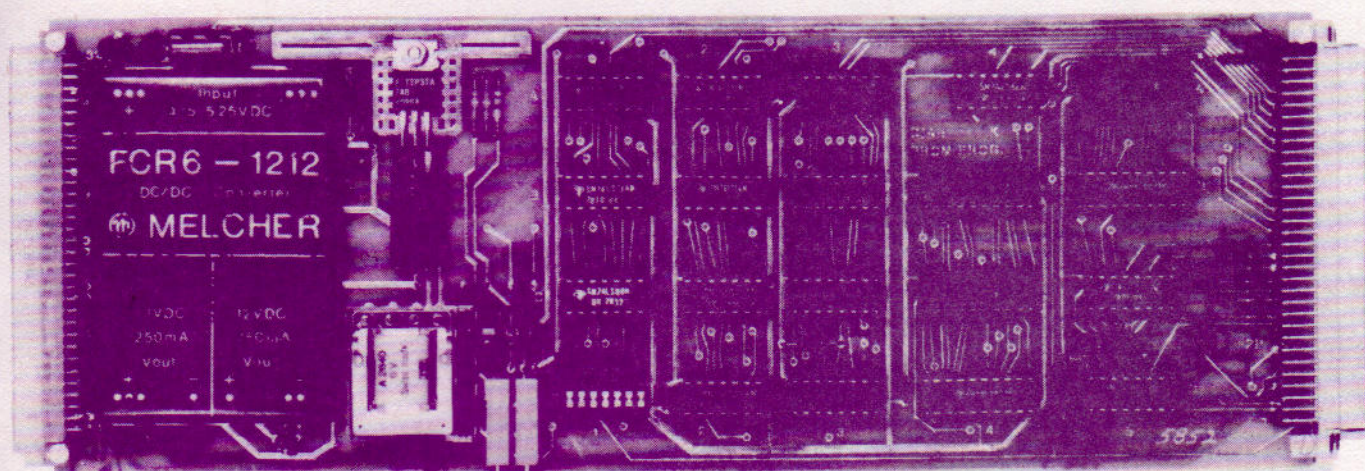
Because of the large capital expenditures involved in the system development it is important that the equipment used is handy and effective. DataBoard 4680 provides standard (8"-diskettes) and mini discette (5"-diskettes) based development systems. See figure 15 and picture 9.

A discette system is more versatile and faster for program development. A program library is easily established and maintained. This gives flexibility and may be used to generate a new program packet by simply linking programs from earlier developments - perhaps with minor changes. System expansion, common corrections or new systems may be developed this way. A new system is then tested with the storage media in a real-time environment. The procedures for corrections and expansions run faster with the advantage of better cost and time schedules. The system designer has the opportunity of checking the system at different stages and yet not losing time as would be the case with a common development system.

9.2 PROM-programmer

DATABOARD 4680 provides a low cost and highly versatile solution to programming requirements for EPROMs type 2758, 2716 and 2732. The PROM-programmer simplifies programming. It minimizes the turn around for reprogramming and revised programs. The PROM-programmer is run on a diskette based development system with the driver and user program(s) on disk files. It is plugged into a spare slot on the I/O-side of the 4680-backplane with the PROM-module plugged into the other end. Memory circuits need not be lifted up from the module, yet the programming is done selectively at chip-level. Operation is conversational. The operator is led step by step through each programming operation. The whole procedure is completed with verification of the PROM-program.

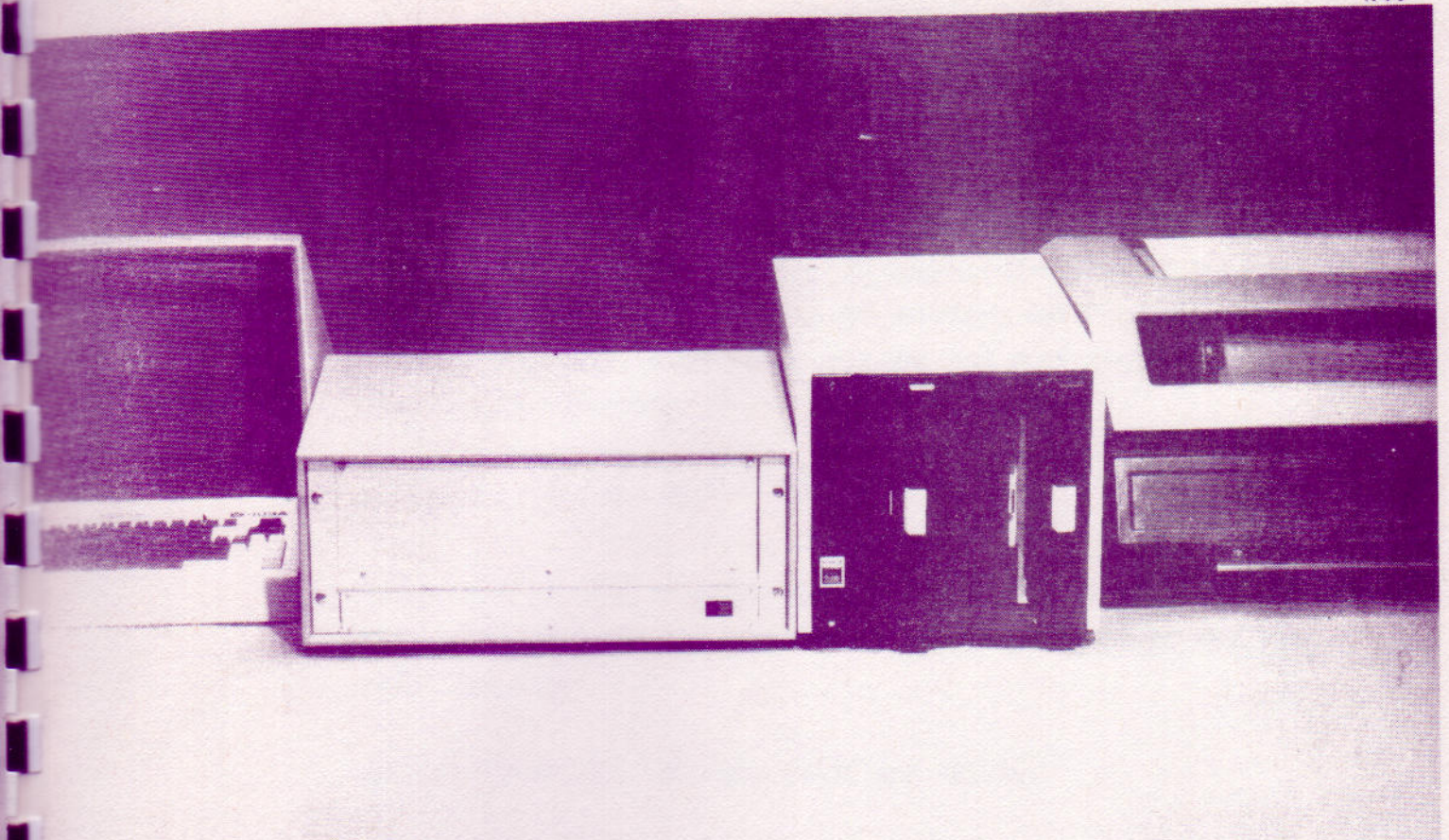
DATABOARD 4680 provides an ULTRA-VIOLET ERASE LIGHT-unit especially suited to its programmer. The eraser has room for a whole module so erasing may be done with the PROMs on-board.



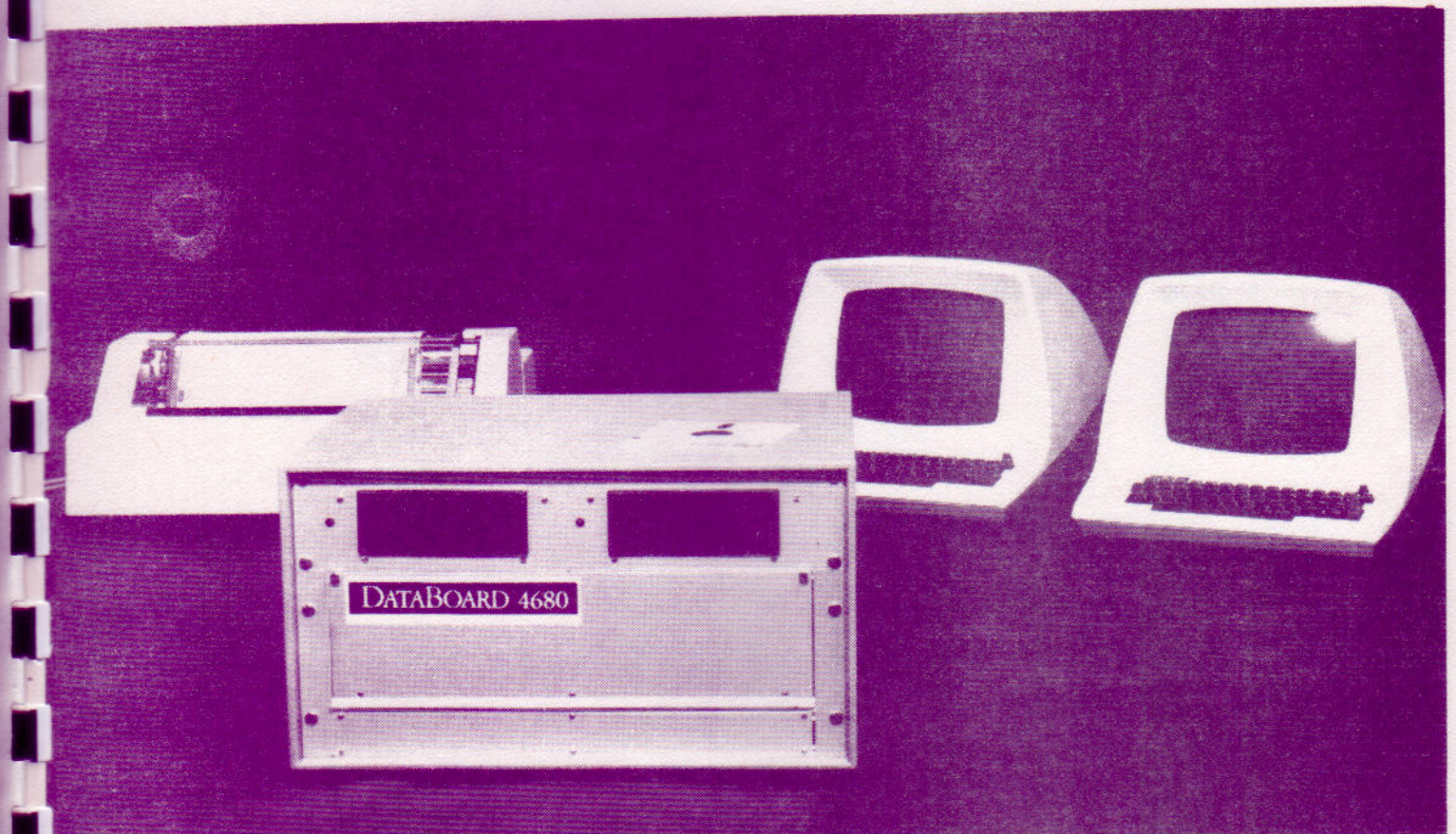
Pic. 8 PROM-programmer

9.3 Programming

The modular build-up technique utilized in the hardware design is also applied to software design. An operating system (OS) is a good starting point. The OS is necessary to move programming fast towards the application part of the system. Common functions as monitoring and program drivers for peripherals are resolved at the beginning. There must be a compact solution for single board-computers to make use of and one that is structured for versatile and powerful use.



Pic. 9a Floppy disk based system.



Pic. 9b Minifloppy based system.

Programming requires following aids (each program is described in chapter 10). The procedure is illustrated by the figure 18.

1. An operating system to provide device and data management for the different development programs. An operating system to run user system programs on a wide range of hardware environments, performing useful system level operations and providing commonly-used general services.

We have DOS and DOS.8MT.

2. Editor for on-line creating and editing of source programs and data files. The user has the opportunity of using assembler or high-level languages.

We have ASSEMBLER, FORTRAN, BASIC and PASCAL.

3. An Assembler to accept source programs written in symbolic micro-processor mnemonics to produce object code.

A Fortran compiler to accept programs written in the fortran language. It compiles source code to produce executable object code.

4. A linker to link together relocatable object code created by the assembler and to produce executable program modules or program packets.

The DataBoard incorporates the linker.

5. A debugger for dynamic trouble-shooting of user programs through a terminal.

The DataBoard provides a very strong debugger facility through DEBUGGER.

6. Loaders for loading programs to memory. Loaders are provided within the respective operating systems. CPU on-board mini-boots are provided on PROMs to facilitate the automatic load of programs at system (re)start or on operator controlled reset.
7. Facilities for PROM-programming.

DataBoard incorporates a card module to be used in an application or development system. It provides a direct path for a program from diskfile(s) to PROM-module. Selective on-board programming at chip-level is provided.

9.4 Development system

A diskette based system contains following components:

- Hardware
1. The backplane and power supply
 2. CPU, the double board computer
 3. 32 kbyte of RAMs (or advisable 64 kbyte)
 4. Floppy-disc, for 5"- or 8"-diskettes, with interface/controller
 5. DMA-processor with the 8"-diskette system
 6. Display unit (CRT) with interface and cable
 7. Printer with interface and cable
 8. PROM-programmer.

This hardware is assembled from standard modules in DataBoard 4680. Picture 9a shows a 8"-diskette system. A 5"-diskette system, shown in picture 9b, is the same with respect to handling and functions but has half the capacity of the other. It works without DMA but has the advantage of being available for development of the single board computers.

Software 9. Disc Operating System - DOS

10. or - DOS.8MT

11. DEBUGGER/MONITOR

12. EDITOR

13. ASSEMBLER

14. and/or FORTRAN

15. and/or BASIC

16. and/or PASCAL

17. LINKER

18. Utility routines

9.5 Peripherals

Databoard provides interfacing to following types of peripherals:

- Teletypes (20 mA)
- CRT:s (V24 or parallel)
- Printers "-"
- Paper Tape Equipment (Facit SP 1)
- Cassettes
- Floppy-discs (or discettes)
Both 5" and 8" discettes are provided
- Video processors in both black/white and colour for direct interfacing of CRT-monitors
- Card readers

Asynchronous/synchronous interfacing for data communication is available.

The IEC-bus is supported by commands in BASIC and provides the user with a standard interface for communicating with instrumentation and other applications using the standard IEEE 488 interface.

10. SOFTWARE

DATABOARD offers a complete set of programs required for program development as described in section 9.3. It provides the operating systems DOS.6 and DOS.8 to service application program development and to act as an environment in which application programs are run. It provides the tools - program development utilities and languages - that enable the user to create executable machine-code to run the application system.

The user is provided manuals on each of these programs to facilitate their functional understanding and operation. Prospective customers may study the DATABOARD software concept in a collection of documents named SOFTWARE CATALOG. It covers each program concerning capabilities, performances and system aspects.

However, a short introduction is made in the following sections:

10.1 EDITOR

This is used by a programmer (normally sitting at a CRT) to write, correct and update source code for programs. The source code is written in any of the provided languages which determine the mnemonics and format to be used. The source code is then saved on to a data file. The user may choose any of the available devices as file saving media. The file is then available for subsequent development work and called in for corrections or based in the development of new programs. The file contents may be listed on a local or remote printer.

10.2 ASSEMBLER

Provides the means for a programmer to create executable object code from a source file containing the mnemonic operation codes - the assembler code. The code complies with the machine instructions of the respective microprocessor used. The editor is used to write the assembler mnemonic loaded on a file. The file is then called in by the assembler to convert the assembler code into object code. The object code is output (loaded) as an object file on to the same device as the source code. The assembler includes the choice of local or remote printer for output listing.

10.3 High-level languages

10.3.1 BASIC

This is one of the most generally used programming languages. It is a fully interactive programming language providing line-at-a-time entry for storage and/or immediate execution. It uses two phases. One for precompiling and the other for execution, allowing for faster program running.

It is structured in a way that the user can take full advantage of terminal functions such as screen and graphics. A multi-user basic is also available.

10.3.2 FORTRAN

FORTRAN is a high level language designed for more effective use by making programming faster and easier. It is a true resident compiler. It is fast and produces highly optimized directly executable code.

It is a subset of ANSI standard FORTRAN IV with powerful extensions for hardware manipulations. Machine code can be included and parts within the CPU can be directly addressed as FORTRAN variables.

10.3.3 PASCAL

This is a high level language independent of any particular machine in order to facilitate the interchange of programs. It is a powerful, flexible and effecient language and particularly strong in its handling of data structures. DataBoard may therefore be used as a powerful tool for writing compilers. For the user it is primarily intended to support data base handling. Interleaved assembler programs will make it flexible for most applications of DataBoard 4680 as process I/O programming.

10.4 LINKER

The linker accepts relocatable object code programs created by the assembler to produce executable program load modules in absolute and binary code.

10.5 DEBUGGER

This is one of the most powerful and useful utilities of the development system. It is used for troubleshooting object code programs. Programs can be tested step by step, or in free run execution including interrupt handling.

A variety of commands are provided for trace functions. They give an effective testing of individual programs and block of programs.

10.6 DOS (DISC Operating System)

DOS is a file management system. It uses overlay technique and has totally dynamic file allocation. It is very easy for an operator to use since all file handling is done by file name. The DOS library maintains 255 files.

The location, expansion and contraction of files, as well as random/ sequential access facilities are automatically provided for by the DDS.

Routines for generating a new DOS system are included. A large set of commands are available to the user.

10.7 DOS.8 (DOS with Multi-Task capabilities)

This is a real time operating system with multi-programming capabilities, and intended to run on a 4680-system based on a Z80-CPU. It can be used in system development as well as in applications. It is designed to meet different types of requirements for real time applications such as process control, production control, plant supervision, communications etc. Single and double board computers are both supported.

Programs can reside permanently in memory or brought into memory for execution from a mass storage device. Execution is carried out on a priority basis. The system has a modular design and only those modules which the user needs have to be included at the time of software installation.

The following indicate useful points in minor applications:

- Task oriented. The programs can be grouped in functional blocks for clean and modular programming.
- System service as calendar, input/output etc is done through special supervisory calls (SVC) with standard parameters.
- Cooperation between different functions (blocks of programs) is performed by standard signals through these TRAPS. Provides stringent software signalling.
- Independence from a physical device configuration is acquired by the logical assignment of devices.
- Peripheral program drivers and handlers are standard.
- General interrupt handling.
- Expansion is easy.

11. I/O-INTERFACING

Through DATABOARD 4680 the user has a large assortment of functional I/O-modules. Following list shows the different groups of interfacing:

- General TTL-interfacing
- High-power drive
- Isolated digital signalling
- Standard interfaces
 - SPI
 - IEC-bus
 - V24/RS232C
 - "20mA current loop"
 - Viewdata/Prestel
- Analog signalling
- Peripherals
 - Paper Tape Eq
 - Cassette Tape
 - Diskette Systems (5" and 8")
 - TTY
 - Fast printers
 - Displays
 - Videoprocessors bl/wh and colour
 - Card readers
- Communication

The user should consider compatibility between the different I/O-modules. Some of the modules are directly interchangeable with respect to programming, for example the modules 4007 and 4009-relay and transistor outputs respectively. The module 4008 can be used to take over the input signalling of the module 4006 if the user requires isolated input. In that case the user codes both cards with the same channel address, and of course, redirects the input field wiring to 4008. Module 4005 can be used to increase the driving capacity of the lower 16 bits of module 4006, the TTL-interface. The modules can be combined by assigning the same channel address to them.

Some of the I/O-modules provide an option for exchanging the standard field interface circuits to a user's special signalling requirements. Module 4005 for example is provided in complement (negative active) signalling.

A sample of standard modules follows:

Digital signalling

4005	2 * 8 buffered digital TTL outputs 1 * 8 digital TTL inputs
4006	4 * 8 digital TTL outputs with ext. tri-state control 2 * 8 digital TTL inputs
4007	8 latched relay outputs, max load 100V/0.5A
4008	16 opto-isolated inputs
4009	8 latched transistor outputs, max load 50V/1A
4088	Interrupt scanner

Analog signalling

4082	A/D-converter	32 * 12 bits or 2 * 12 bits
4083	D/A-converter	4 * 8 bits
4084	D/A-converter	2 * 12 bits
4089	Flying capacitor	Provides high common and isolation MPX for A/D-conversion, 10 channels

Peripherals etc

4017	UART	RS232C/V24 or current loop interface
4073	"	(4017). Interfaces serial transfer
4087	"	peripherals as teletypes, CRT:s, modems etc. Data rates of 50 to 9600 baud.
4026	USRT	Communication V24/RS232C
4038	Fast Printers	Centronic
4060	" "	CDC
4037	Card Reader	CDC
4034	Diskettes	8"-diskette interface
4076	"	5"-diskette interface/controller
4025	IEC-bus	
2066	Video processor	Black/white
2081		Colour for Viewdata/Prestel text
4015	SPI-SEND	Parallel interfacing of
4016	SPI-RECEIVE	FACIT-equipment

12. CONFIGURING MEMORY

The 4680-system provides a large number of different modules based on (E)(P)ROM, static RAM and dynamic RAM. External capacity is obtained through diskettes (standard or mini), cassettes or paper tape.

A dividing line distinguishes the double board computer from the single board computer according to external memory and the use of mass storage devices. The DMA-function is provided with the double board computer. DMA facilitates the use of faster mass storage devices as standard diskette. However, DataBoard 4680 provides a good and less expensive alternative by means of the mini diskette available for all 4680-systems.

Minor systems are normally based on (E)(P)ROM and use the on-board capacity (8 kbyte PROM and 1 kbyte RAM) of the single board computer.

Static RAM with battery back-up is available for use in systems that require data saving at power-off etc.

RAMs on the other hand need a loading device for programs from which programs are loaded at system start and restart. A restart or bootstrap program is essential. DataBoard 4680 provides a method for this with the double board computer. CPU on-board restart programs and "watch-dog" facilitate the keeping of memory intact.

The choice between the different RAM-modules is generally a matter of whether to use DMA or not. DMA demands a limited time for memory access (specified in the data sheets of the DMA-processor). The use of slow memories is provided by the double board computers.

When choosing between static and dynamic RAMs the following general differences must be considered:

- dynamic memories run cooler than statics and, since they can be packed more tightly together, save on cost and board.
- static memories use + 5 V only.

Memory configuring requires nothing more normally than the selection of the address area and the plugging in of the module. The address selection (card selection) is dealt with in figure 20.

Figure 16 shows the expansion steps provided through module and module capacity.

A sample of RAM- and PROM-modules is demonstrated as follows:

2 K RAM (2002)	Static RAM based on 21L02B On-board expansion in steps of 1 K
4/8 K RAM (2055)	Static RAM based on 2114L-3 On-board expansion in steps of 1 K
16 K RAM (2056)	Dynamic RAM based on 4116-3 Supplied with full memory capacity mounted
1 K CMOS RAM (2027)	Static CMOS RAM based on 5101. Battery back-up for 100h. Automatic recharge. Programmed memory protection.
8 K EPROM (3032)	Based on 2708. On-board expansion in steps of 1 K.
8/16/32 K EPROM (3061)	Based on memory circuits 2758, 2716 or 2732. On-board expansion in steps of 1, 2 or 4 K depending on the selected memory circuit. Facilitates on-board (re)programming by the PROM-programmer 8071.

13. POWER REQUIREMENTS

AC power requirements are:

$220V \pm 10\%$, 50 ± 1 Hz

Option for other line voltages is provided.

A fully configured 14-slot rack dissipates maximum 125W.

The user calculates actual dc power requirements by obtaining the load of each individual module from respective data sheets. The appropriate power supply module(s) is then selected from DATABOARD 4680 product line. Be sure also that the load of the options is summed up.

DATABOARD power supplies maintain a constant output voltage in spite of changes in input voltage, output loading or temperature. The regulation requirements must be met if the user wants to make his own choice of power supply and are as follows:

+5 V

Stability at net variations, $\pm 10\%$:	0,01%
Ripple:	< 50 mV
Regulation time:	< 1 msec
Overvoltage protection must be provided.	

± 12 V

Varations for load variation 0-100%:	1,5 mV
Stability at net variations, $\pm 10\%$:	0,1 mV
Regulation time for load variation 0-100%:	15 μ sec
Ripple:	0,6 mV

14. POWER SUPPLIES

Standard power supply modules are available in three levels:

- 1) for the 8-slot Single Board Computer backplanes
- 2) for the 14-slot backplane
- 3) for the 20-slot backplane

All power supplies are selected to meet the power requirements of any respective installation. The user should always check the dc requirements when configuring a system. The dc load of each module is found in data sheets of the module.

14-slot power supply

One set comprises plug-in modules for rack mounting to be connected to the net via a standard 4680-transformer. This power supply set has enough capacity to drive a fully mounted 14-slot rack. The capacity is:

-	+5 V	10 A	product number	7904
-	±12 V	1 A	product number	7905
-	transformer		product number	7903

20-slot power supply

This power supply is self-contained and connected directly to the net. The module is placed in the inner back of boxes. The capacity is:

-	+5V	20 A	product number	7907
-	+12 V	1,75 A	included	
-	-12 V	1,75 A	included	

8-slot power supply

The supply is included in the hardware assembly product number 7908.

-	+5V	5 A		
-	+12V	0,3 A		
-	-12V	0,3 A		

15. ASSEMBLING THE SYSTEM HARDWARE

The 4680-system is a modular design which facilitates installation and service. The user may easily put together the different modules to make a complete system. Additional hardware for expansion is easily introduced.

The customers have the option of assembling systems by themselves or ordering complete ready-to-use systems.

DATABOARD offers development systems with the basic hardware assembled and run through a complete functional test. These box-contained systems comprise a 14-slot backplane mounted on a 19"-rack, power supply, a double board computer, a bootstrap loader, a mass storage device and basic amount of memory. The systems are :

8900 + 7900 Development system based on 8"-diskette storage.

8901 Development system based on 5"-diskette storage.

15.1 Backplane

The 4680-bus is provided as a backplane to which the modules are mounted as plug-in boards. The backplane is available in standard racks for 19" rack mounting or for box enclosure. The user can choose from standard 8, 14 or 20 slot racks. The 14 slot rack contains slots for the power supply as well. The box mounting has room for the power supply in the inner back of the box. CPU is placed in a fixed slot. The other modules may be placed in the optional slots of the respective bus-sides. The special signalling of the DMA-processor and UART are prewired into the standard backplanes and are therefore provided with fixed slots. See figure 13 and picture 5a and 5b. The user is not obliged to use the standard racks. The backplane wiring may be done by the user himself. Expansion is provided by the bus-expansion card; an expansion module which includes a cable to the next rack.

The standard backplane is keyed to prohibit the unintentional turning of the wrong end or side of the card to the bus. The user has at the same time the possibility of keying his external cable connectors so that the external signals are connected to the right card.

Forced cooling is recommended for the 14/20-slot racks.

The 4680-bus backplane is available as follows:

<u>Product Number</u>	<u>Description</u>
7901	14-slot backplane mounted on a 19"-rack. The slots are assigned: 2 for power, 7 for I/O, 2 for CPU and 5 for memory.
7902	20-slot backplane mounted on a 19"-rack. The slots are assigned: 13 for I/O, 2 for CPU and 5 for memory.
7908	A box-contained 8-slot backplane for single board computers. Power supply is included. The slots are assigned: 4 for I/O, 1 for CPU and 3 for memory.
7918	A box-contained backplane for ABC-80 expansion. The slots are assigned: 4 for I/O and 2 for memory. Power is included.

For bus-expansion:

5023	BUS-expansion card with cable.
------	--------------------------------

15.2 Power

When you connect the power supplies to the backplane check for correct connection: remove all cards first, then turn the power on and measure.

An appropriate power cable and plug is supplied with all box-contained systems. Note that a ground pin on the plug must be connected to the normal service ground to ensure safe operation.

15.3 Interrupt signal wiring

The interrupt signals are prewired into the standard 14/20-slot back-planes. See figure 6. INT 1 - INT 7 are wired in successive order to the seven first slots. INT 0 is left so that the user has the highest level for free choice. Figure 11 lists recommended interrupt levels (IL..) for standard I/O-devices.

Figure 18 shows an imaginary configuration with its interrupt wiring. The demand for response states the priority level. The unit on interrupt level 0 has the highest priority and will therefore be serviced prior to the others and therefore interrupts any job running on a lower level.

Several units may be connected to the same interrupt level. Bits in the status-register can be used to identify the individual interrupts on the same level. The disable/enable function can also be used to separate interrupts so that only one is active at a time.

15.4 Card modules

All card modules in the 4680-system are of the European standard dimensions:
100 x 160 mm

The connector type used is B64 pin DIN 41612 on the bus-side and the same applies to the other side for I/O-signalling. The power supply modules for the 14 slot rack use 31 pin DIN 41617.

Special interfacing, such as video signalling, requires special connectors.

15.5 Configuring memory modules

When assembling the main memory the user will be concerned with:

- address selection
- mounting PROMs (and RAMs). RAM-modules are normally shipped with full capacity mounted.
- access time
- wait-state signalling

Address selection is done with an address selection plug or switches on-board. Figure 20 describes the principal method. Respective module data sheets show the code and location of switches or code plug.

The location of the memory circuits is also provided in the data sheets.

Access time is dependent on whether DMA is used or not.

If DMA is used, select the RAM module(s) and memory circuits to match DMA access time requirements. Wait-state signalling is of no significance for DMA. See DMA data sheets.

If no DMA is used, you have a free choice of RAM-modules. Check the CPU access time requirements. Single Board Computers give you no choice. You have to select modules and circuits that are fast enough to match the CPU. In the case of Single Board Computers it does not matter if the module jumper for wait-state signalling is installed or not. The jumper is installed as standard and is cut if no wait-signalling is required. If your system is based on the Double Board Computer check which optional wait-state signalling applies in your memory configuration. The wait-state signalling is fixed on the dynamic RAM-modules. The dynamic RAM-modules require the wait-state signalling at instr. fetch (not optional). See sections 4.2.3 Access times and 4.2.4 Wait-state signalling.

15.6 Configuring I/O-modules

The normal steps in configuring I/O-modules into the systems need present no difficulties. The steps are as follows:

1. Select the channel address; usually a code plug. The data sheets show the on-board location. The coding is described in figure 20.
2. Select jumper-selectable options if any. Most of the options are wired on the connector of the I/O-cable.
3. Select the position for the respective cards. Check that the interrupt signalling is wired and that it fits.
4. Position the module into the backplane.
5. Plug the I/O-cable and bring the cable through the bottom (in 14- and 20-slot racks) of the rack to the backside and then to your own screw terminal strip for further field wiring.

Standard peripherals are provided with ready-to-use cables which connect the device to the interface module.

The way to insert modules may seem obvious but damage can still be caused. Consider the following points:

1. Never insert or remove modules when power is applied.
2. Make sure you are inserting the module in the right bus-side.
3. Position the module correctly, right side up. You may succeed in forcing the field side of an I/O-module into the backplane.

Relay operations may generate crosstalk interference which adversely affects A/D-conversion. Such interference is avoided by installing the relay and A/D modules separate from each other by other I/O-modules in between.

Large systems may need expanded capacity for I/O interfacing. The standard backplanes have room for 7 I/O-modules in the 14-slot version and 13 I/O-modules in the 20-slot version. Expansion is accomplished by wiring the I/O-bus on an expansion rack. The bus-expansion card, including cable, provides the required bus-drivers and connection. The I/O-modules are prepared for backplane expansion by the CSB-signal (described earlier in this document).

The secondary channel selection (CSS*) is normally available on the I/O-bus.

Its purpose is to service user designed expansion of the I/O-functions of a particular card. The CSS-signal from the mother card will in these cases service the channel selection for the daughter cards. If CSS-signal is used make the wiring between the concerned cards.

The user should keep a complete record of the details of the entire system configuration. This is particularly important with respect to the optional jumper-selected functions and differences from the standard design.

15.7 Spares holding

The company provides spares holding for at least one of each of the functional module used in any system, and offers a service contract for repairs on an exchange basis for faulty modules.

16. A GUIDE FOR CONFIGURING A SYSTEM

1. Select I/O-modules for your signal interface.
2. If you have special interfacing requirement then design your own I/O-module by using the Prototype-card with the bus interface included.
3. Select memory for program and data. You have at your disposal:
 - (E)(P)ROM
 - static RAM
 - dynamic RAM
 - static RAM with battery back-up.

Static RAM requires +5 V only. Dynamic RAM requires less boards. Both require a loading device if used for program memory.

4. Select loading device if you have program in RAM. The following devices are available:
 - floppy-disc standard
 - " mini
 - cassette
 - paper tape equipment
5. Check the application against the environment and start from the beginning if there are obstacles.

The system can be distributed into different levels. Parts of the system may be based on the single board computer and placed local to an interface signal. It is connected to the main system by serial communication. The serial interface contributes flexibility to the application. Peripherals can be placed in better and more accessible environments. System availability and service are other important factors that are improved.

6. If the system incorporates standard floppy then add in the DMA-processor too.
7. Do you need data communication? Add in USART too. A special slot is used in the backplane.

8. Which operator communication is required? Select among:
- Teletypes
 - Console typewriters
 - Printers
 - CRT:s

The UART-module is used to interface to these peripherals.

Videoprocessor is available for direct interfacing to a CRT-monitor

- black/white video
- colour video

9. Is real-time clock required?

The single board computer provides on-board CTC. Special I/O-module is available.

10. Is the system specified for precautions at "Power-Down"? If so then design the required logic by yourself and connect the interrupt signal to the NMI, the highest and nonmaskable interrupt. All bus signals are shown in figure 6.
11. Does the system incorporate hardware control of the program flow and scheduling? This is provided by the Watch-Dog in both single and double board computers.
12. Is restart or mini-boot specified for your system? The double board computers have separate PROM:s for it. You have the possibility to use standard mini-boots or to design the program (1 kbyte) yourself.
13. You get automatic I/O-trace through hardware as option in the double board computer. The function is specified in data sheets of the double board computer and also in chapter 6.
14. Sum up the required processor functions and select the computer - the double board or the single board computer. Refer to chapter 3.

15. List the spare modules required.
16. Count the number of modules on the memory - and I/O-sides of the backplane. Select the rack. You have at your disposal 8, 14 or 20-slot standard racks. The number of slots may be expanded with an expansion card and cable to the next rack. These are provided as system parts in DataBoard 4680.
17. Make a scheme of the used slots. See figure 13.
18. Sum up the necessary power and select the power modules from DataBoard 4680.
19. The rack can be mounted in standard box from DataBoard 4680. The backplate and fan is provided in the 19" box. This backplate is prepared for mounting the transformer in the case of power card modules.
20. Check the backplane on the optional wiring. The recommended interrupt levels are shown in a table in figure 11. INT 0 in the standard backplanes is left for user own choice.
21. If I/O makes use of wait-states then the appropriate CSB-signals must be connected to the IORDY (available in the Z80 double board computer).
22. Make an overall check of the "wait-states" on the memory-side. See "configuring memory". The MEMRDY signal is jumper-selected, normally placed on the address-plug. With the Z80 double board computer you must select one of the alternative conditions for wait-states on the CPU-board. The standard is "wait on instruction fetch". Normally you don't need any wait-states.
23. List all cards separated in I/O- and memory-side.

Allocate in the 4680 memory map the volumes of the different memory cards and select the address codes by switches or jumpers on the respective cards. The multiples of memory stepped by the code is specified in the data sheets of each type of memory card.

Select the I/O channel addresses. Recommended addresses (the system channels) are presented in figure 11.

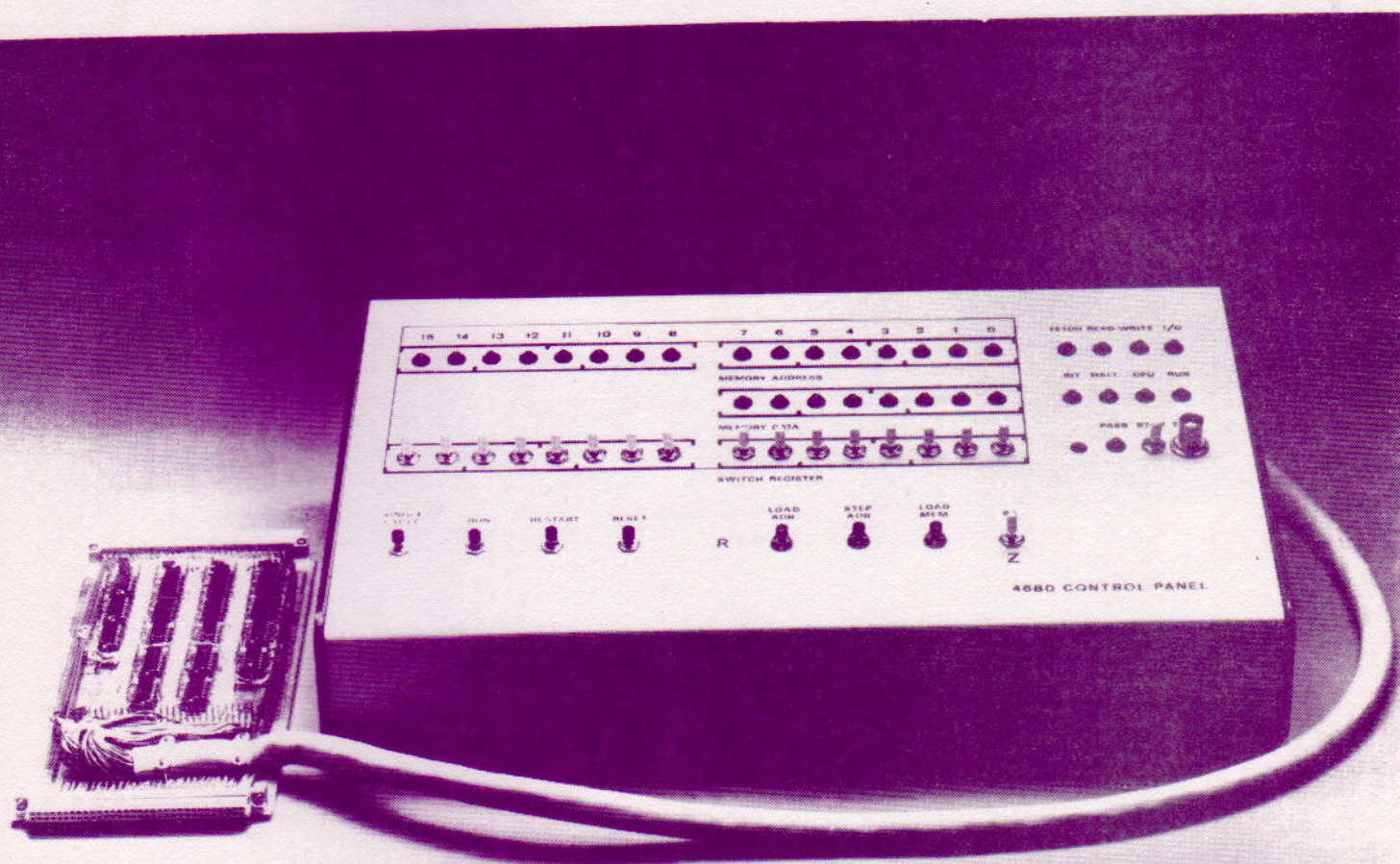
Cut the code into the code-plugs and mount the plugs on the respective cards. Figure 20 deals with the codes.

24. Is restart program incorporated? If so select the volume on the Control Unit of the double board computer. See the data sheets. The standard mini-boot is mounted on 1/2 K.
25. Load the PROMs and/or the loading device with the system program.
26. Switch on the CPU on-board restart program if you have it incorporated.
27. Start the system by "reset".
28. Connect the Watch-Dog.

17. TEST FACILITIES

17.1 Control Panel

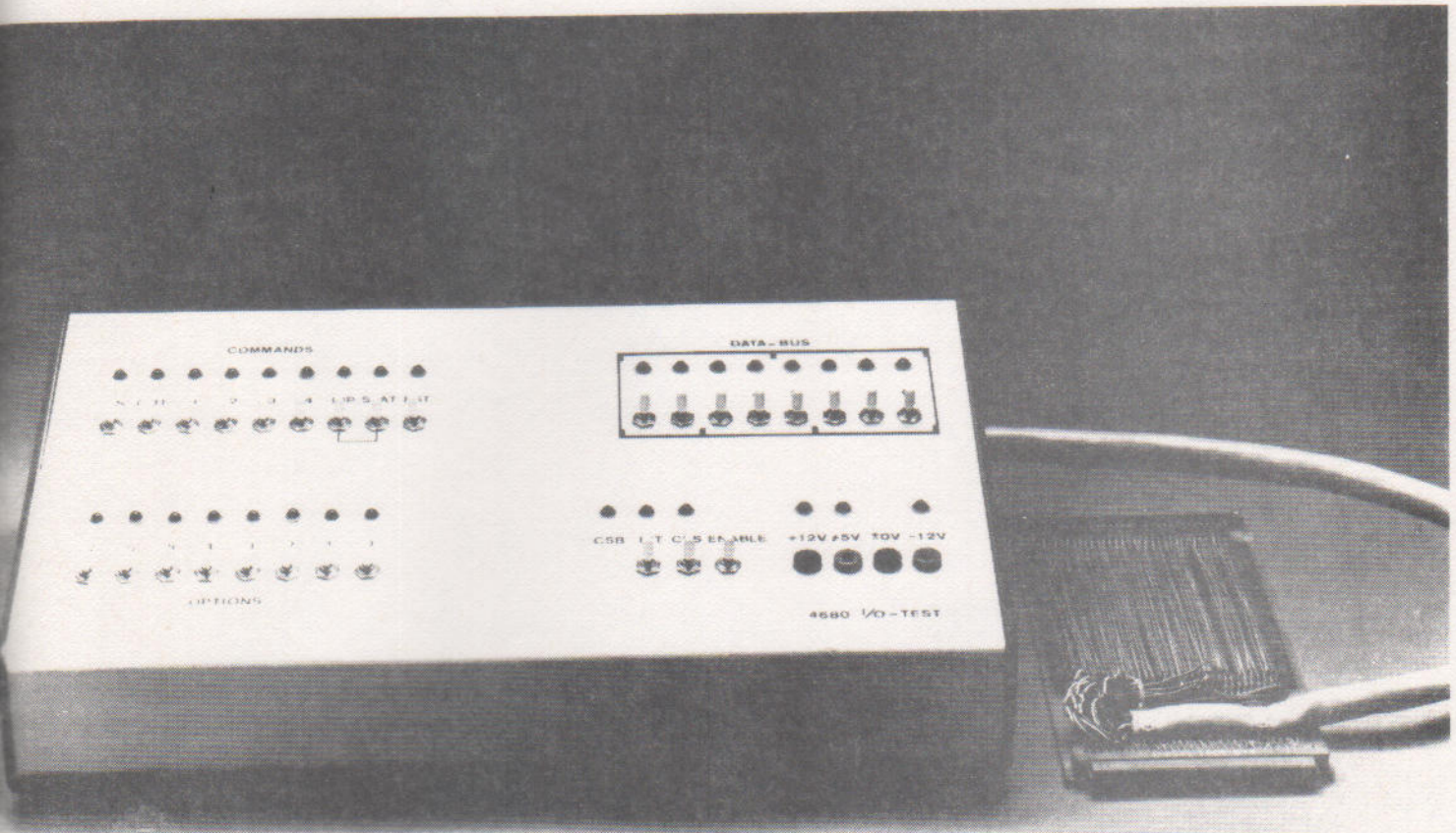
The Control Panel, is as the name indicates, a panel for controlling processor functions. It provides equivalent test facilities as the "Debugger" on a hardware level. It is valuable for quick and on the spot troubleshooting in real signal environment. The interrupt signal handling can be overviewed without delaying software support. All system parts can be held intact during the test procedures. Only the system software and hardware belonging to the application is active.



Pic. 10 DataBoard 4680 control panel.

17.2 The I/O-tester

The I/O-tester provides possibilities to simulate the interface between the CPU and the user process. I/O-faults can be isolated as external or internal to the computer system.



Pic. 11 DataBoard 4680 I/O-tester

17.3 Testprograms

Hardware testprograms are provided. They are delivered on the same media as the development system but can also be loaded on PROMs cassettes etc.

Cassettes are flexible and provide for the user's own special tests. The equipment comprises the cassette drive, I/O-interface module, bootstrap program and a set of cassettes for the respective tests.

17.4 In-built test facilities

Every I/O-module provides a LED on the front-side of the rack to indicate that channel's selection.

18. TRAINING

18.1 Off-shore training

Courses are regularly offered.

A 2-days' course for new users covers system theory, hardware software and hands-on laboratory exercises.

A 2-days' course is also available for more experienced DATABOARD 4680 users.

The students are handed a complete set of DATABOARD manuals and other instruction materials. Please send for the current schedule and detailed information about courses.

18.2 On-site training

Customers may find it more economical to conduct courses at the user's site. On-site instruction eliminates travel and other expenses incurred by students attending courses at training centers. This method allows DATABOARD instructors to emphasize points of particular value to the student's application and operations.

19. ORDERING FROM DATABOARD

In general ordering is very simple as far as the customer system specification is defined. Placement of an order by a customer can be simple or complex, depending on the size of the order, the type of equipment and the extent to which custom and excise specifications must be registered.

A sales ordering list containing all DATABOARD 4680 products is available for prospective customers. The list presents each product by product number and price. When placing an order always refer to the product number and provide information regarding optional functions. Most of the products are described in more detail in data sheets, which also include information about optional features.

Make sure to distinguish between shipping destination and billing address, if different.

The products listed in the price list distributed to prospective customers applies also as a list of off-the-shelf products.

Figure

1. CPU and 4680-bus
2. Single Board Computer and bussignals
3. Double Board Computer and bus
4. Single Board Computer and bus
5. Bussignals and Cardfunctions
6. Backplane for Double Board Computer
7. Interfacing Single to Double Board Comp. Backplane
8. Bussignalling on I/O-interface
9. Sequence of signals at card selection
and following commands
10. Function of CSB-signal
11. Table on reserved I/O-CHANNEL ADDRESSES
(CS) for standard devices
12. Table on reserved I/O-ports
13. Front-view of 14/20-slot backplane
14. System Development Components
15. Development Systems
16. Memory - int. vectors and modules
17. DOS and RTE-structures
18. Building an executable program
19. Wrapping in interrupt
20. Code Plugs

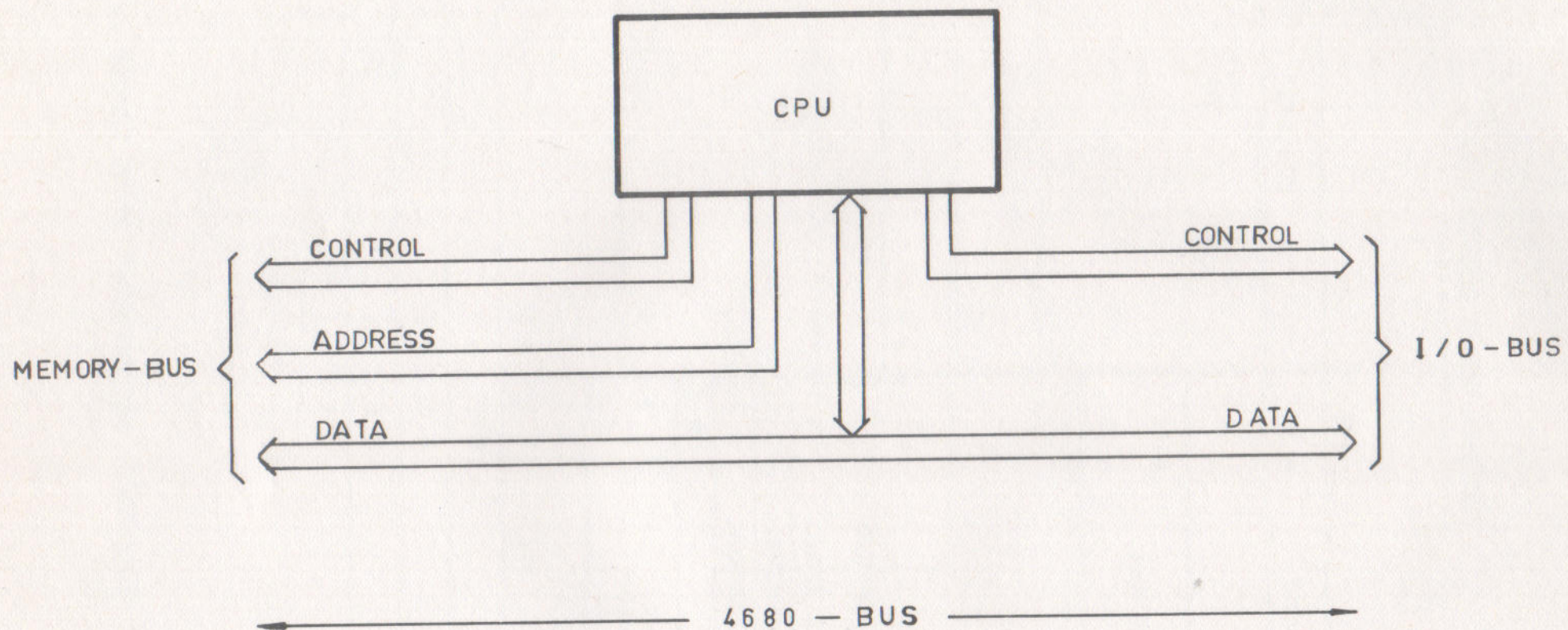


FIG. 1 CPU and 4680 - BUS

MEMCRY BUS
=====

SINGLE BOARD COMPUTER
=====

I/O-BUS
=====

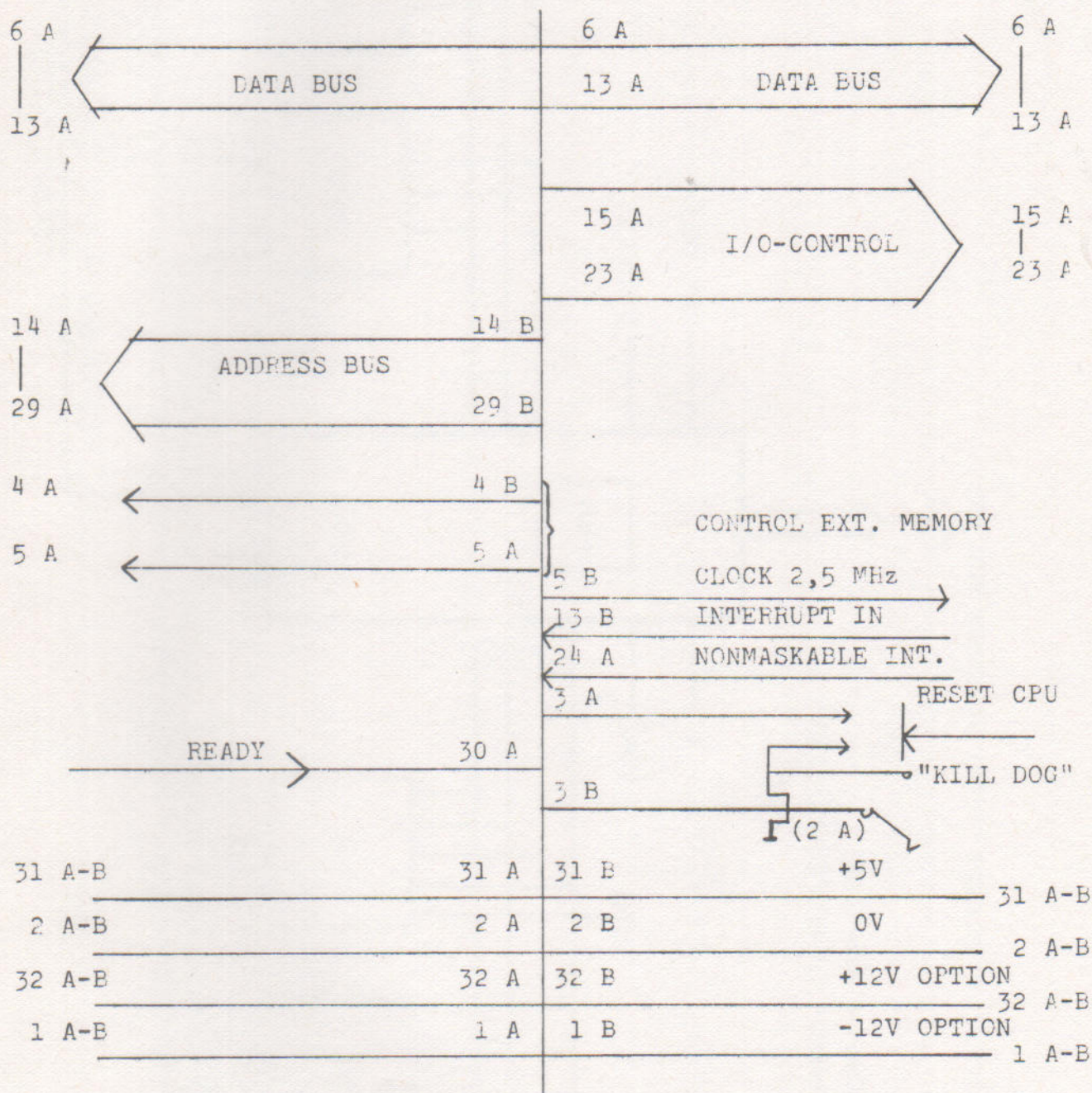
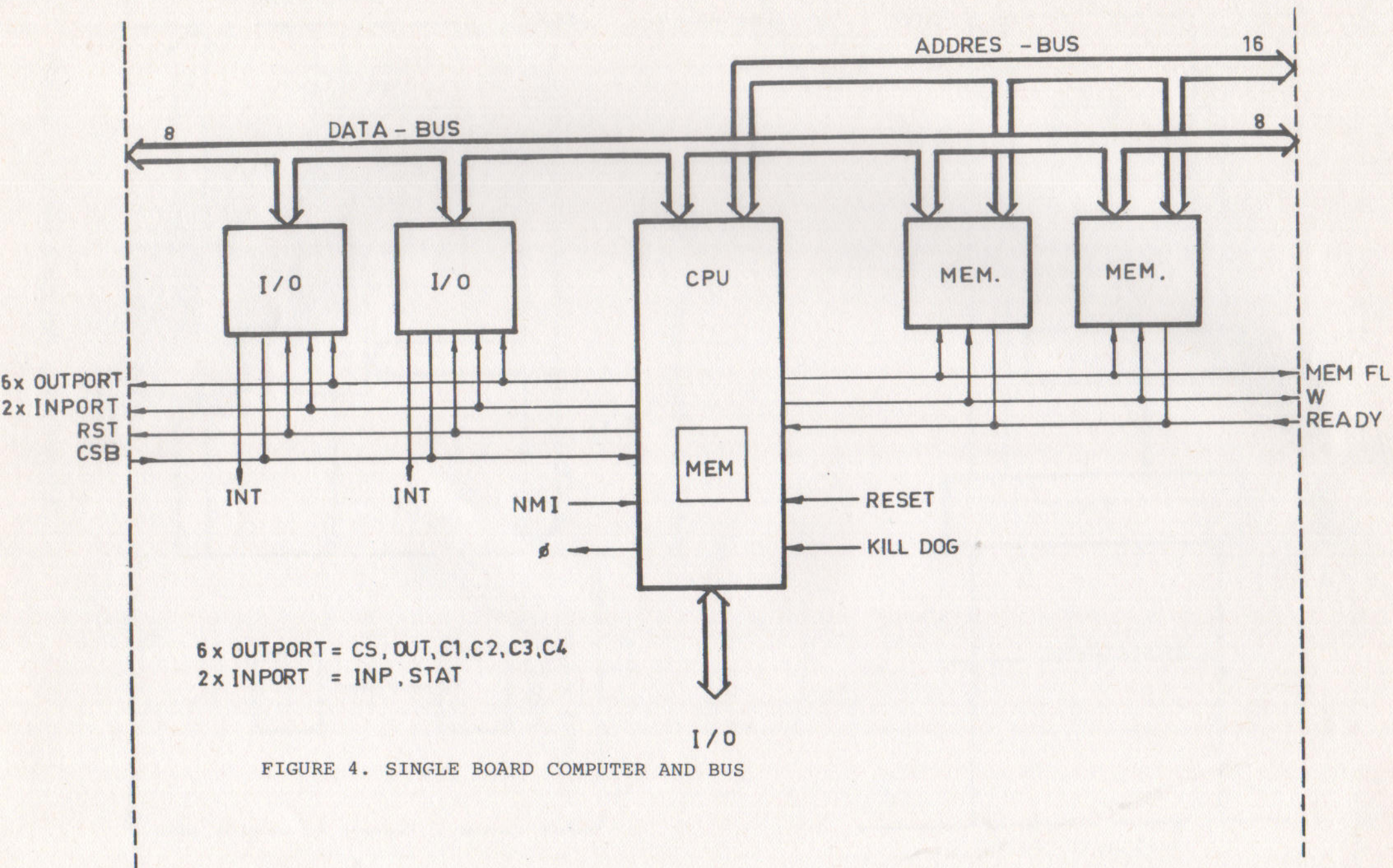
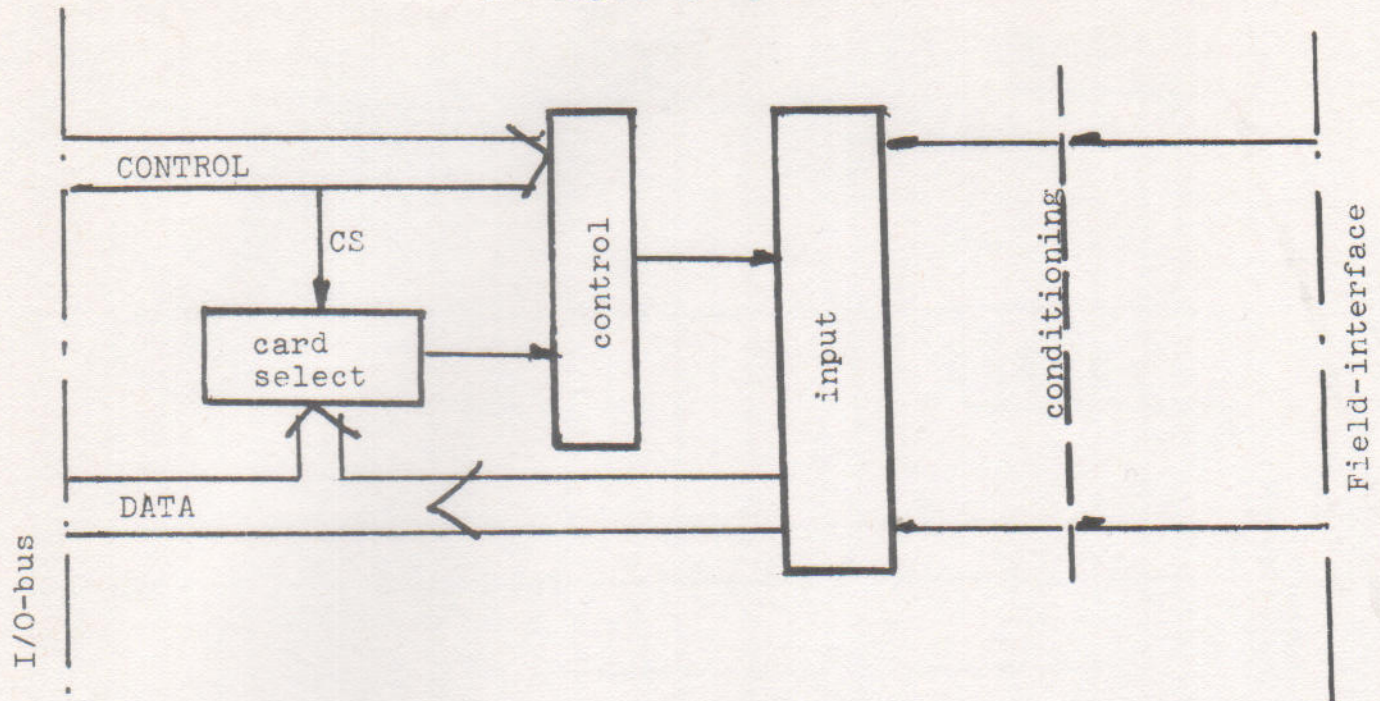


FIGURE 2. SINGLE BOARD COMPUTER AND BUSSIGNALS



I/O input-card, functions



Memorycard, functions.

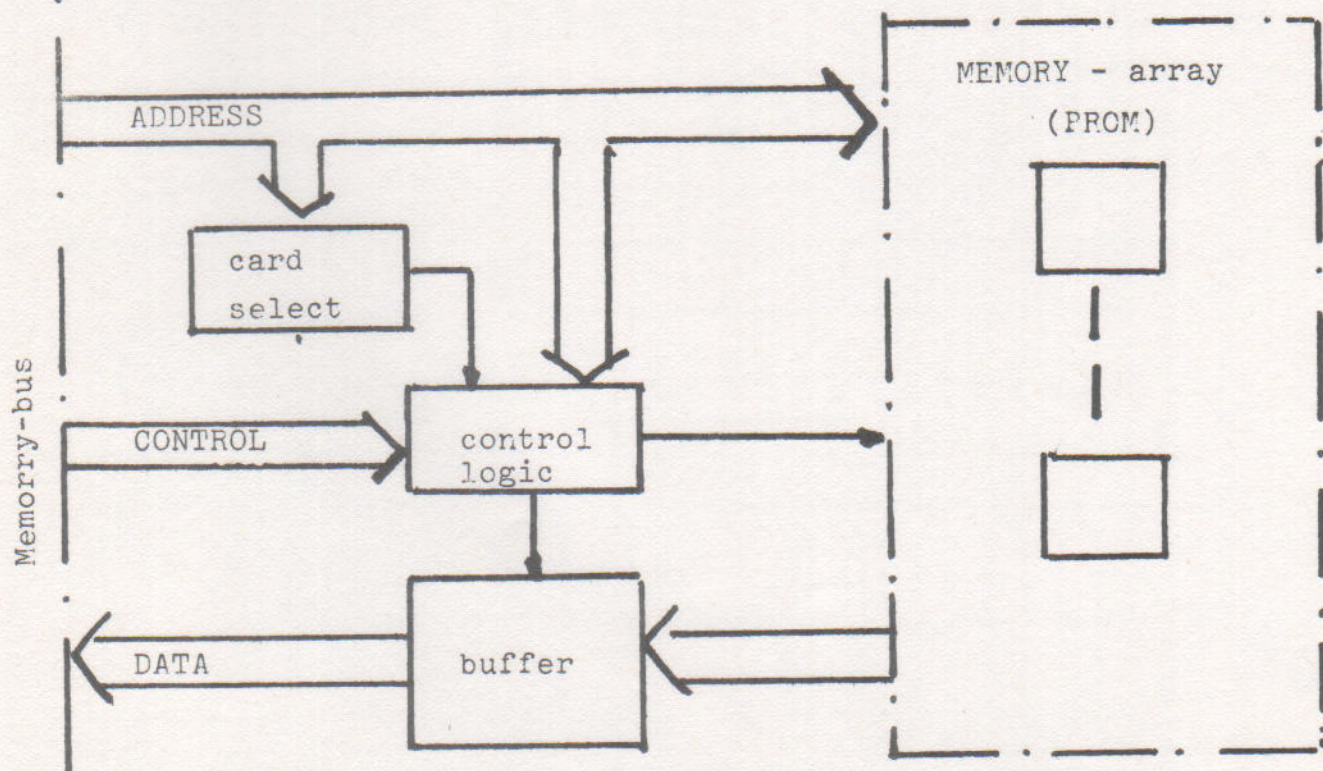


FIGURE 5. BUSSIGNALS AND CARDFUNCTIONS

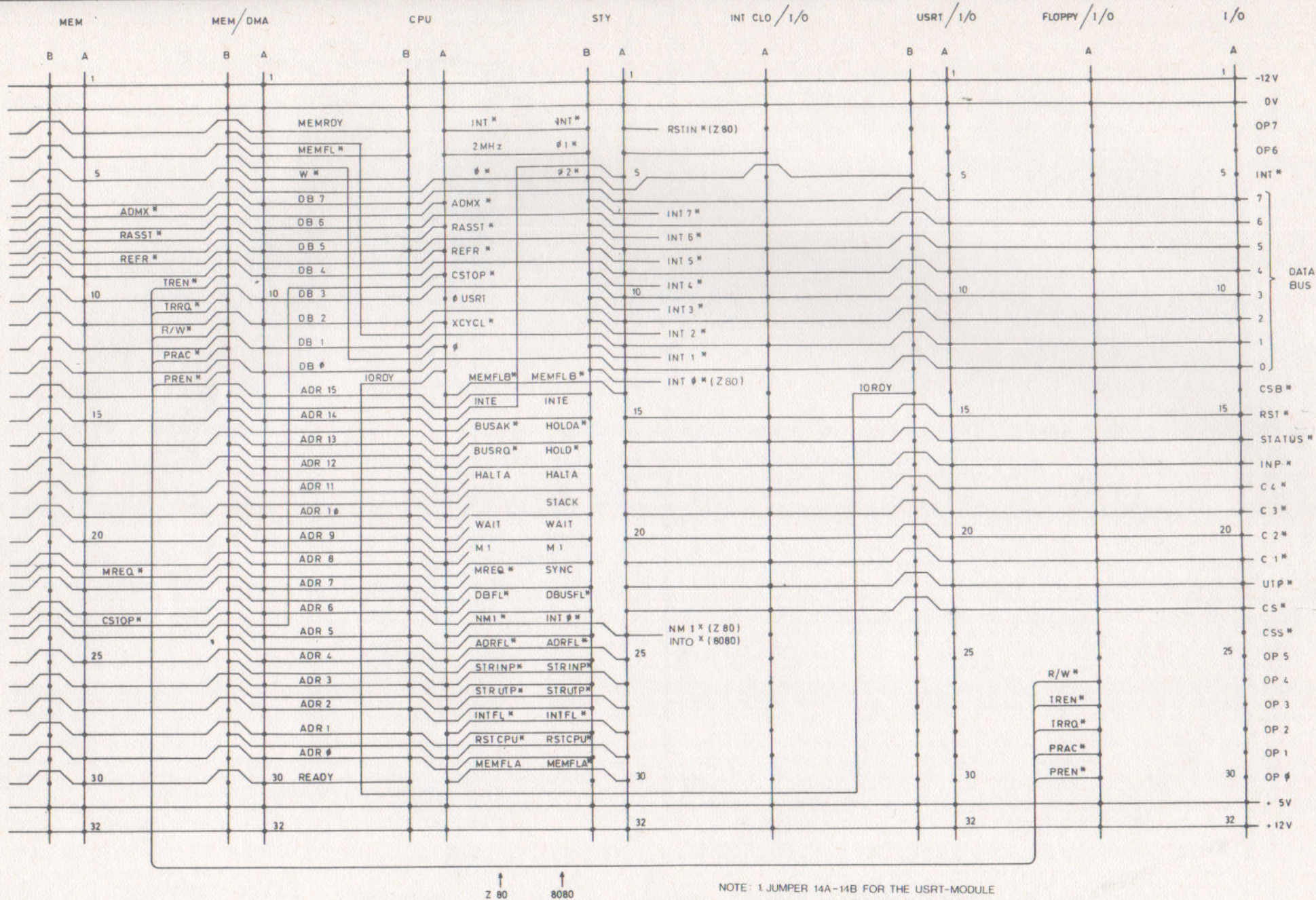


FIGURE 6. BACKPLANE FOR DOUBLE BOARD COMPUTER

		CORR NR	DATAINDUSTRIER AB	BACKPANEL	4680 B 24
		DATE	TABY SWEDEN		

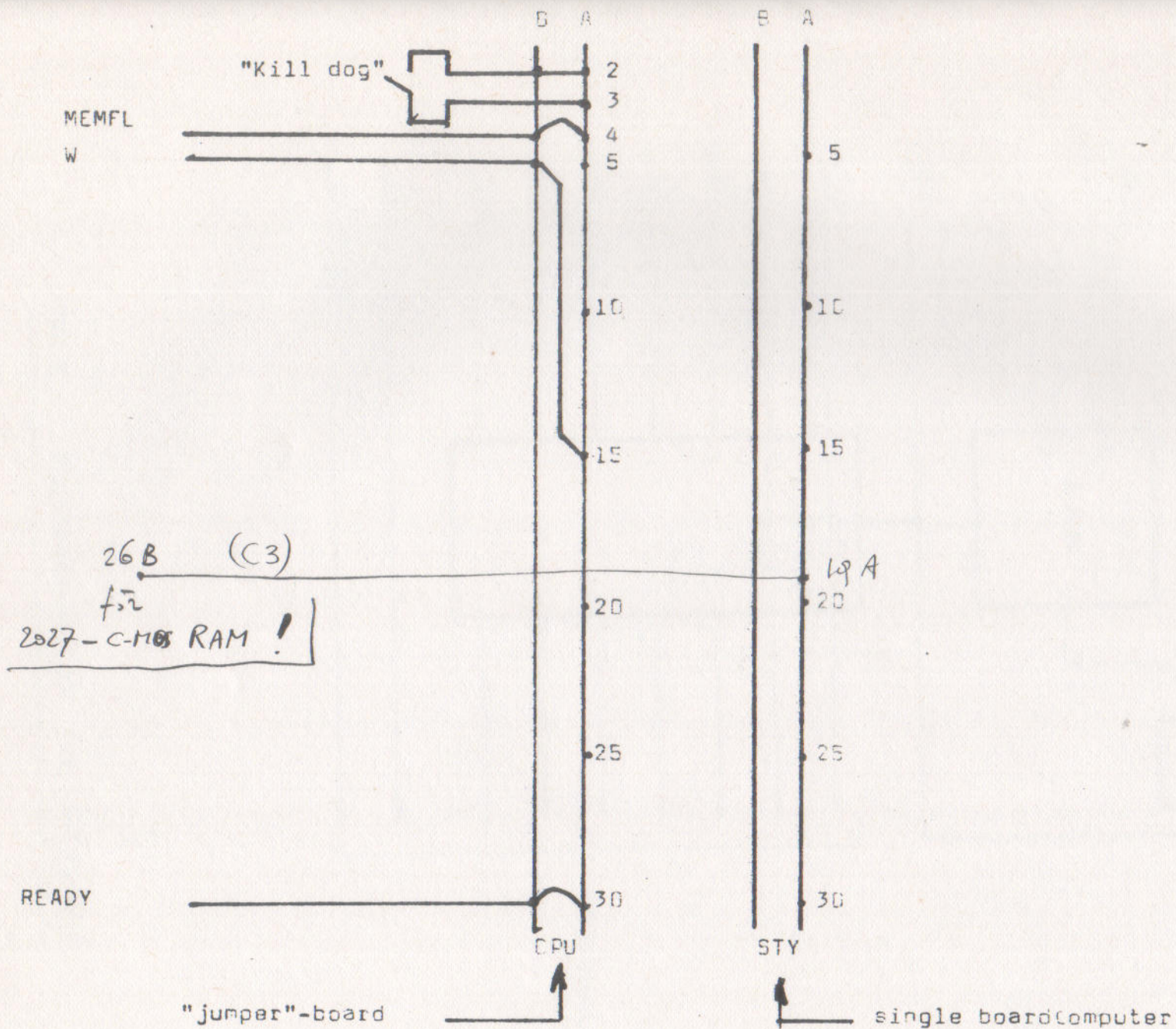


FIGURE 7. INTERFACING SINGLE- TO DOUBLE BOARD COMPUTER, BACKPLANE

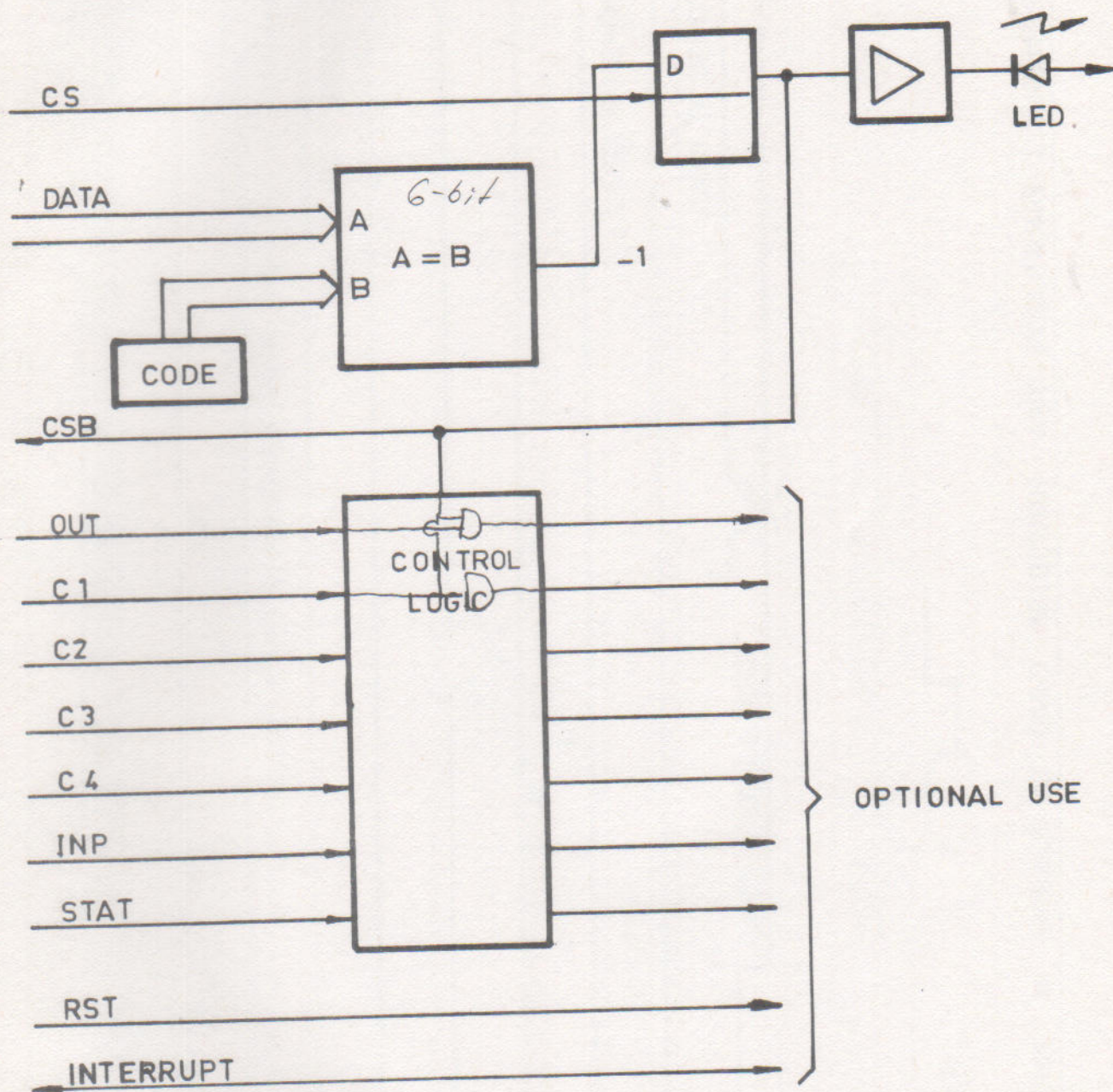
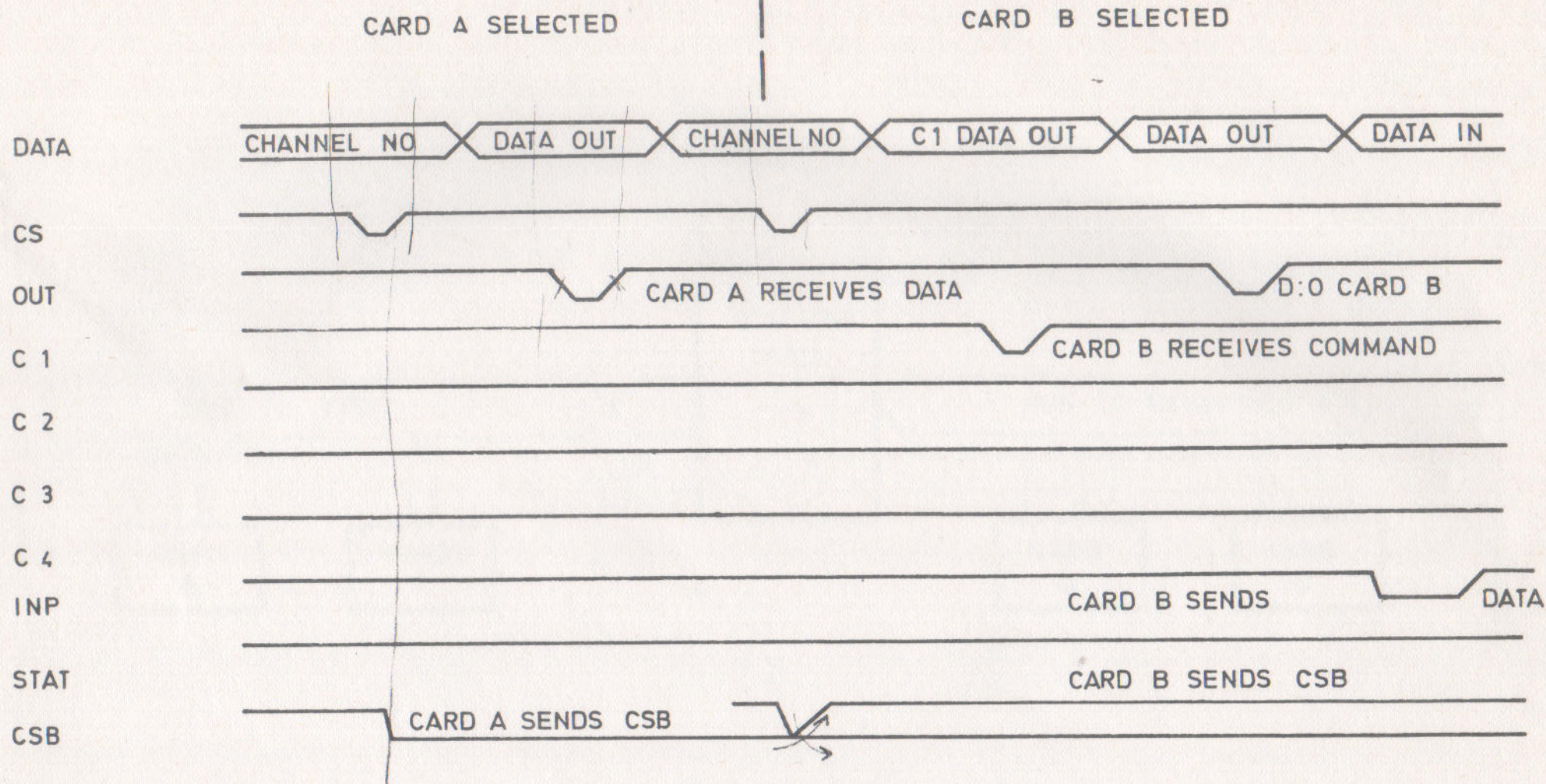


FIG. 8 BUSSIGNALLING ON I/O INTERFACE



ANM. LOGIKKONVENTION:

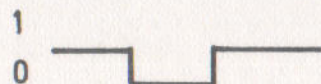


FIGURE 9. SEQUENCE OF SIGNALS AT CARD SELECTION AND FOLLOWING COMMANDS

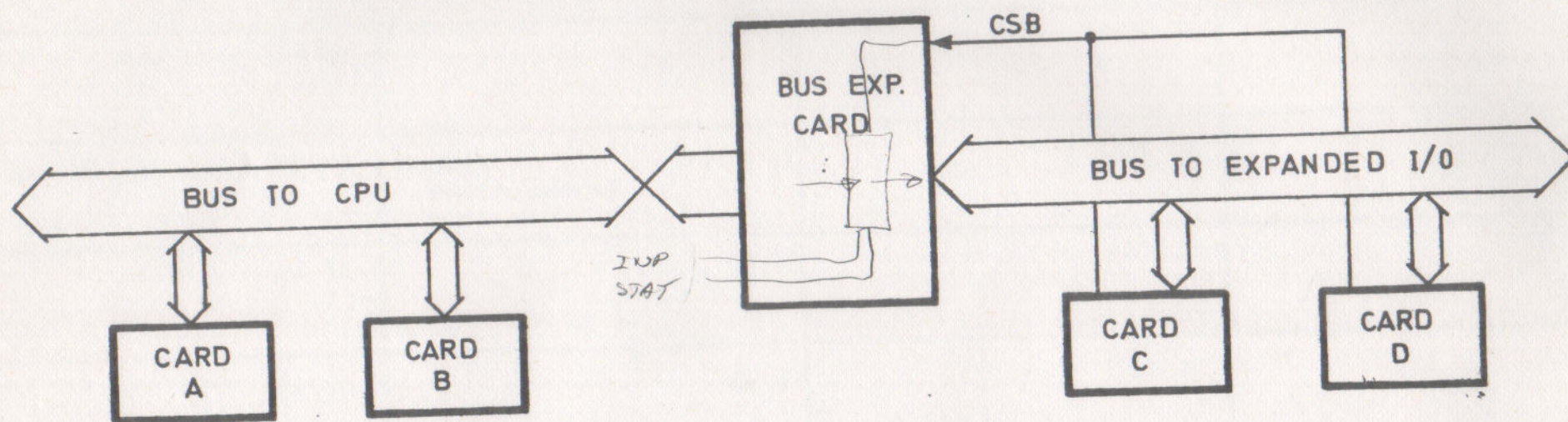


FIGURE 10. FUNCTION OF CSB — SIGNAL

For special application dependent I/O use Channel 1 to 7 and then Channel 61 to 67.
 Avoid using any reserved I/O Channels. When writing new programs use the standard
 devices as far as possible.

Intervall Clock	CS 00 IL 1	CS 01	CS 02	CS 03
	CS 04	CS 05	CS 06	CS 07
Prom Programmer	CS 10	USART (Batch users) CS 11 IL 3	CS 12	CS 13
	CS 14	CS 15	CS 16	CS 17
	CS 20	CS 21	CS 22	CS 23
	CS 24	CS 25	CS 26	CS 27
	CS 30	CS 31	CS 32	CS 33
	CS 34	CS 35	CS 36	CS 37

FIGURE 11a. RESERVED I/O-CHANNEL ADDRESSES (CS) FOR STANDARD DEVICES

Flexible Disc 1 Unit 0-6	CS 40 IL 6	Flexible Disk 2 Unit 7-11	CS 41 IL 6	512 kb Drum 1	CS 42 IL 6	512 kb Drum 2	CS 43 IL 6
5 Mb Moving head disc 1	CS 44 IL 6	5 Mb Moving head disc 2	CS 45 IL 6	10 Mb Moving head disc 1	CS 46 IL 6	10 Mb Moving head disc 2	CS 47 IL 6
FACIT 4203 Cassette tape (ECMA 34) Unit 1 input	CS50 IL2	FACIT 4203 Cassette tape unit (ECMA 34) Unit 1 output	CS51 IL2	FACIT 4203 Cassette tape unit (ECMA 34) Unit 2 input	CS52 IL2	FACIT 4203 Cassette tape unit (ECMA 34) Unit 2 output	CS53 IL2
3M Mini Cassette Unit 0-3	CS54 IL2	Mini Flexible Disc Unit 0-2	CS55 IL6	9 Chann. Mag tape Unit 1	CS56 IL2 6	9 Chann Mag tape Unit 2	CS 57 IL 2 6
Controll panel for Batch terminal users	CS60		CS 61		CS 62		CS 63
	CS 64		CS 65		CS 66		CS 67
Card reader (CDC) 300 or 600 CPM	CS 70 IL 2		CS 71		CS 72		CS 73
System printer SP 1	CS 74 IL 5	System Console UART	CS 75 IL 4	System paper tape punch SP1	CS 76 IL 5	System paper tape reader SP1	CS 77 IL 5

FIGURE 11b. RESERVED I/O-CHANNEL ADDRESSES (CS) FOR STANDARD DEVICES cont.

IN-PORTS

INP 0	}	Read data from selected channel
INP 1		Read status from selected channel
INP 5		Connect restartprogram on the wanted unit (STY)
INP 6		Reset watch dog
INP 7		Reset I/O System
INP 8		Status from DMA

OUT-PORTS

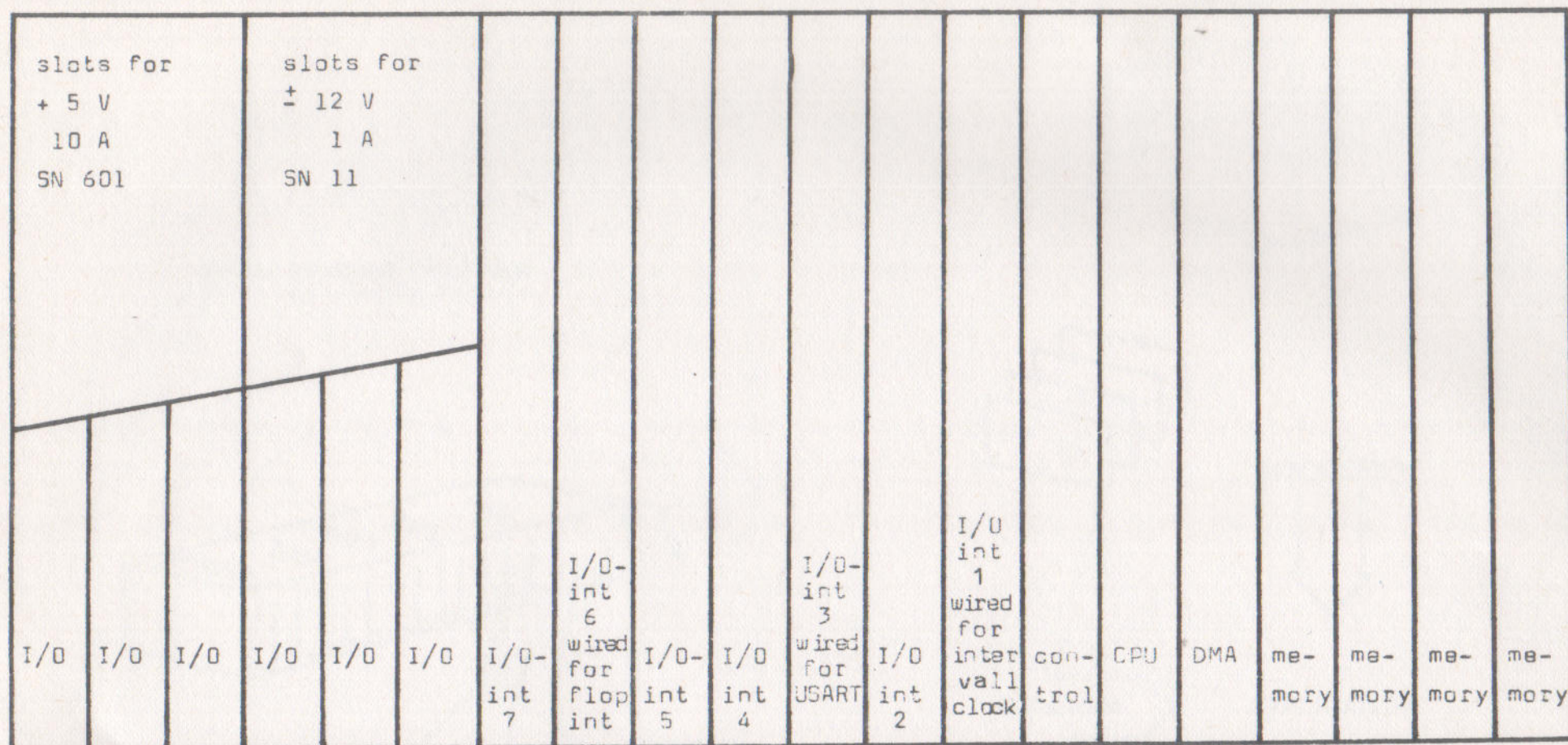
OUT 0	}	Data to selected channel	
OUT 1		Channel selection	<u>key</u>
OUT 2	}	C 1 I/O Command 1	
OUT 3		C 2 I/O Command 2	
OUT 4		C 3 I/O Command 3	
OUT 5		C 4 I/O Command 4	
OUT 6		Disconnect restartprogram and return to ordinary programarea	
OUT 7		Load interrupt-mask	
OUT 8	}	Adr	7 8
OUT 9			
OUT 10	}	LOAD DMA CONTROLLER	size
OUT 11			

CONVENTIONS:

C 4 for disable/enable of interrupt
Data bit 7 = 1 for enable
 = 0 for disable

FIGURE 12. TABLE ON RESERVED I/O-PORTS

Occupied by
power in 14-slot backplane



Occupied by I/O in
20-slot backplane

FIGURE 13. FRONT-VIEW OF 14/20-SLOT BACKPLANE

DEVELOPMENT SYSTEM

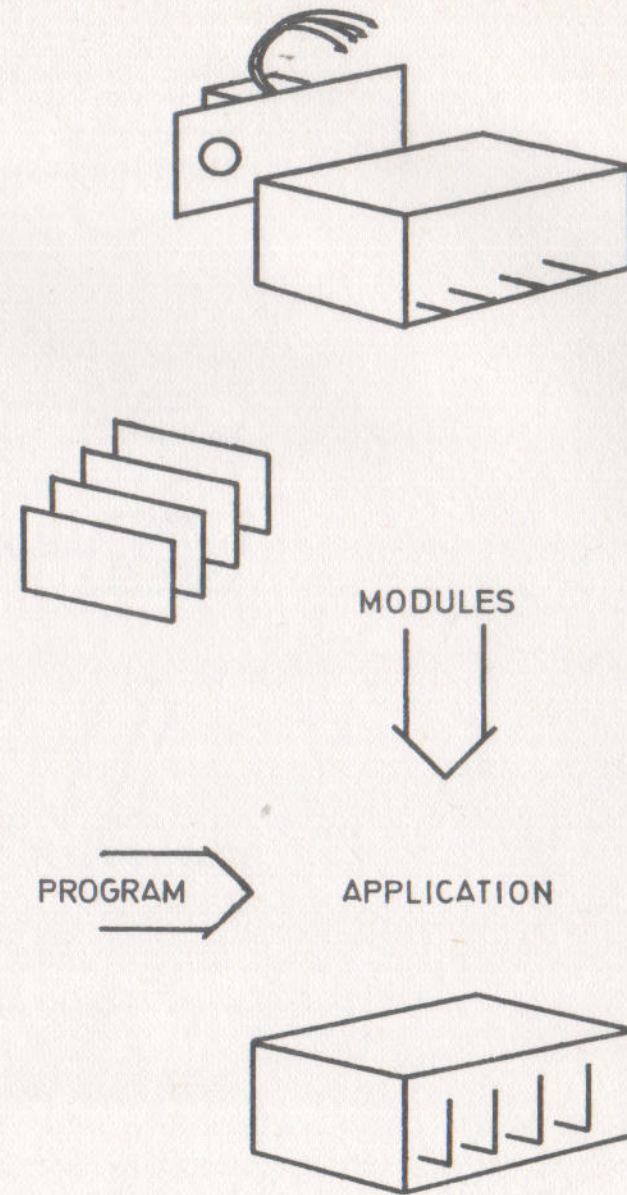
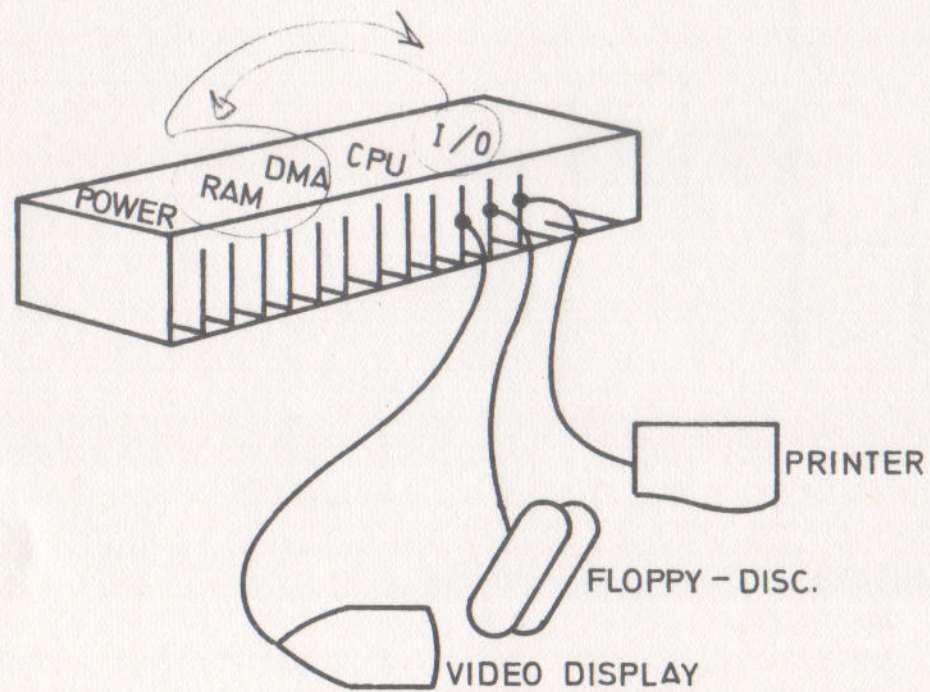
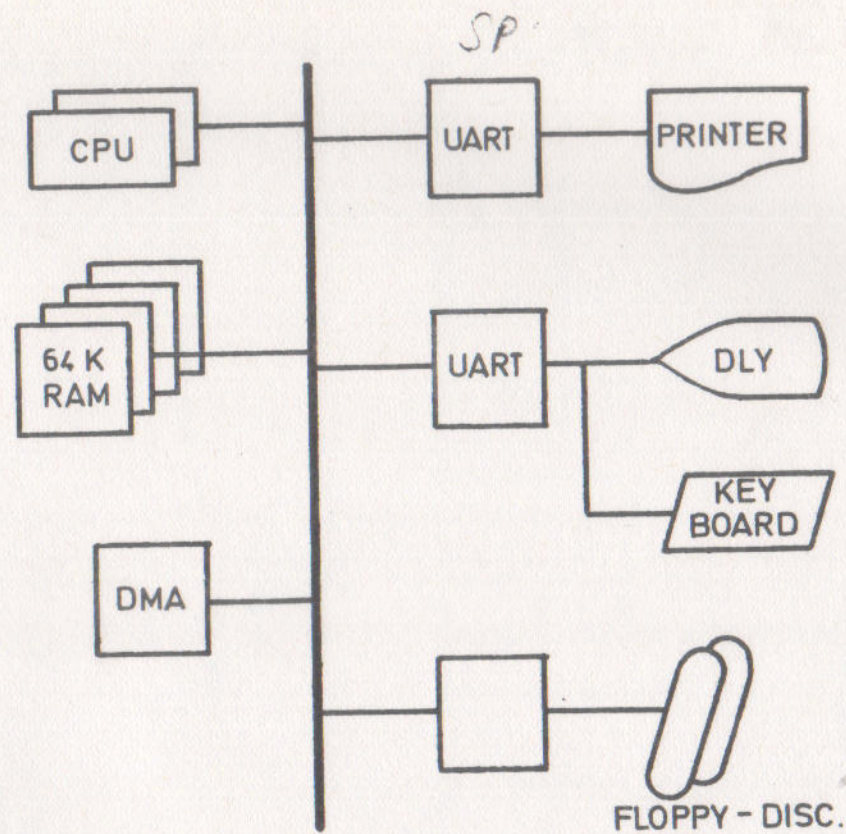
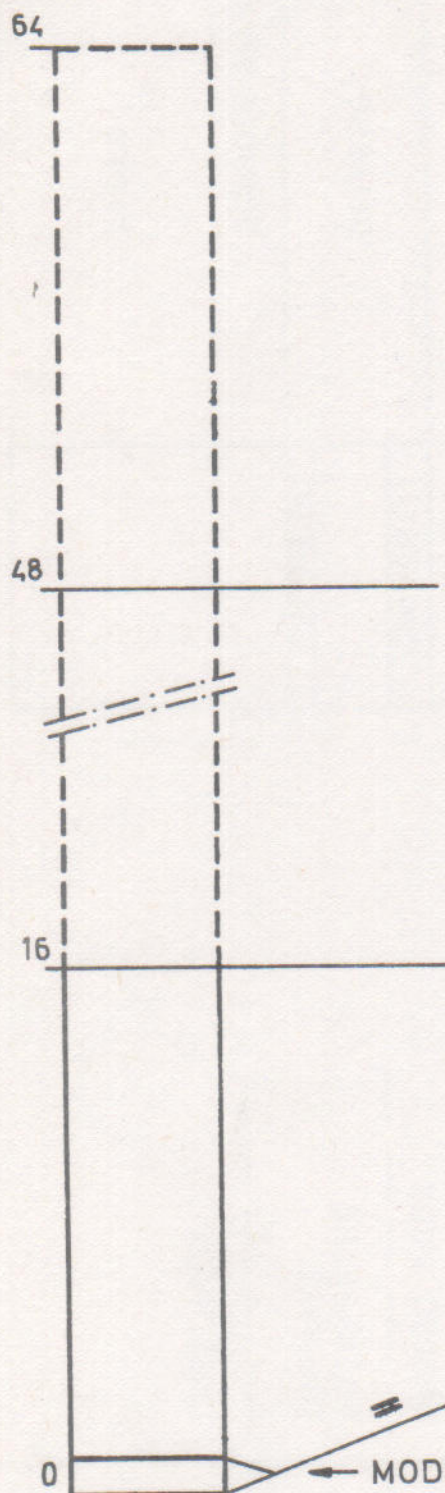


FIGURE 14. SYSTEM DEVELOPMENT COMPONENTS.



FLOPPY - DISC. BASED SYSTEM FOR DEVELOPMENT

FIGURE 15 DEVELOPMENT SYSTEM.



MEMORY BUILT UP IN MULTIPLES OF:

- (P) ROM 1, 2, 4, 8, 16 AND 32 K
- STAT. RAM: 1, 2 AND 8 K
- DYN RAM: 16 K

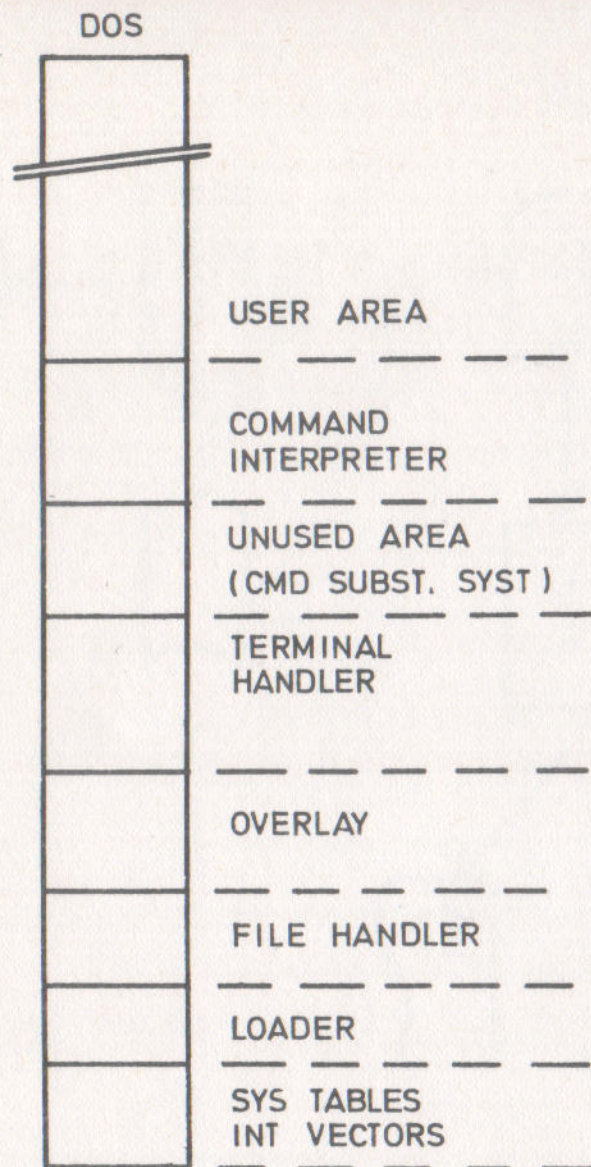
MODE 2

ALLOCATION FREE IN
MULTIPLES OF 256 BYTES

256 BYTE - VECTOR - AREA
FOR INTERRUPT HANDLING.

?

FIGURE 16 MEMORY INT. VECTORS AND MODULES.



DOS. 8 MT

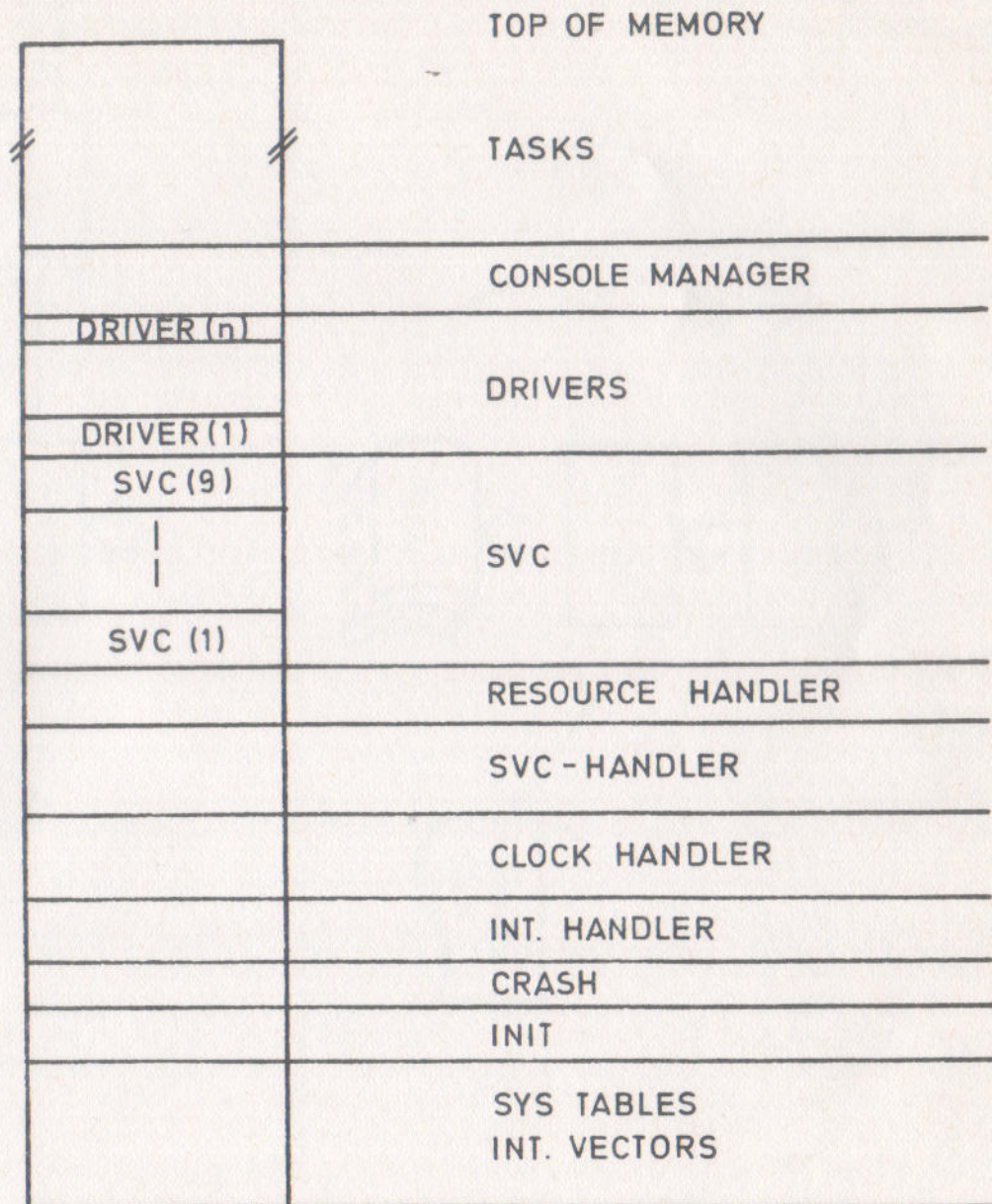


FIGURE 17 DOS.6 - AND DOS.8 MT - STRUCTURES

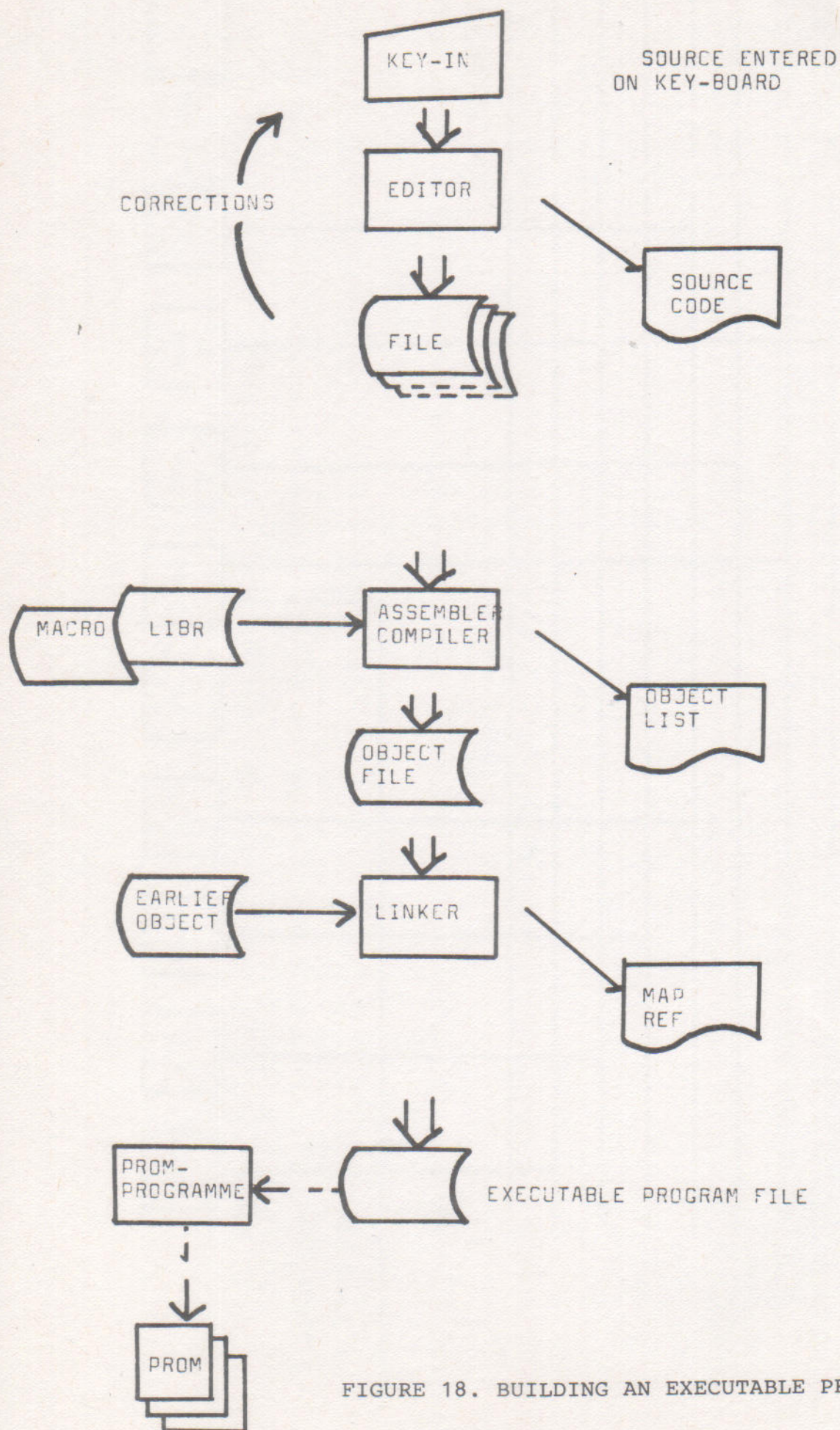


FIGURE 18. BUILDING AN EXECUTABLE PROGRAM

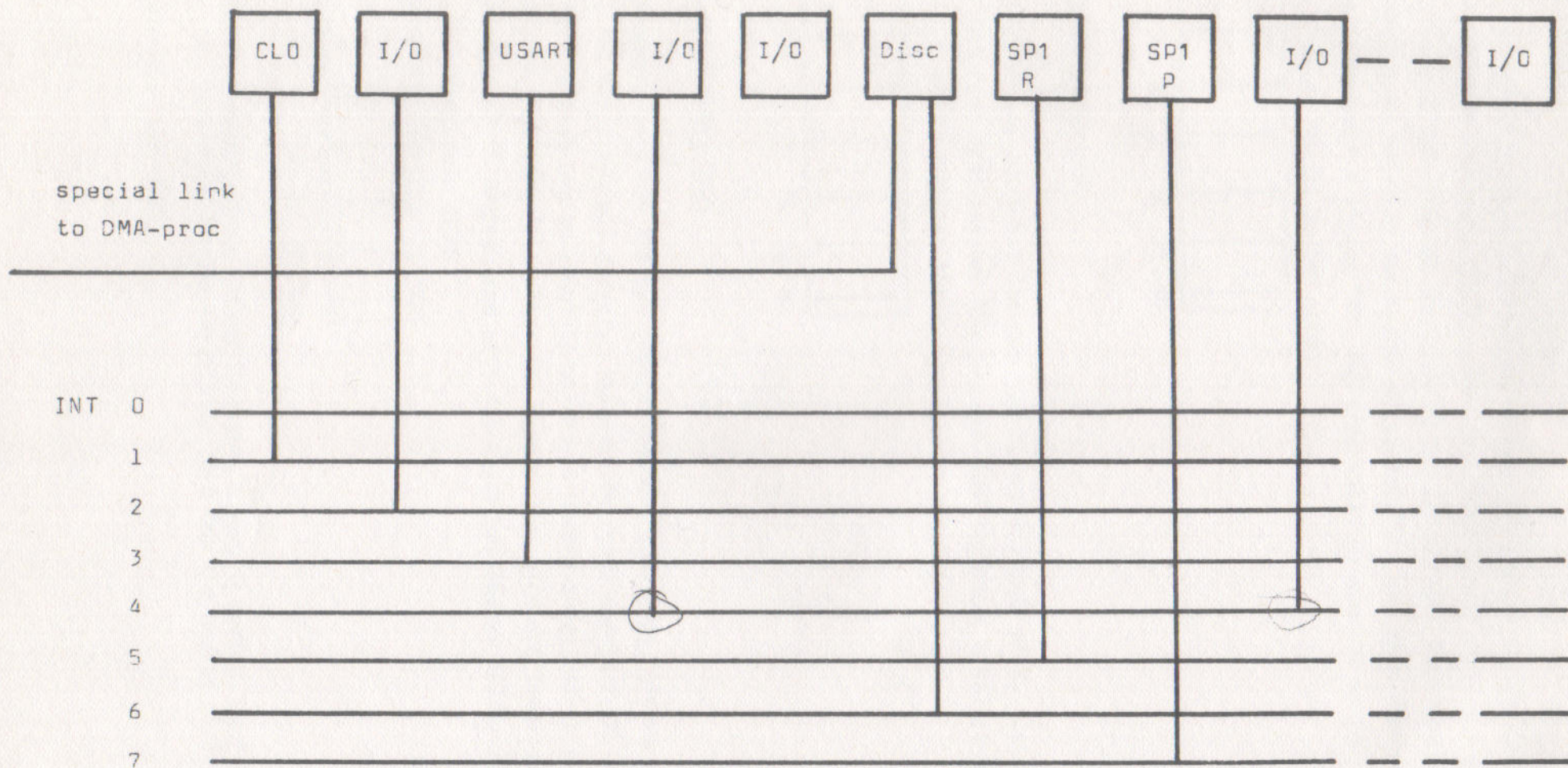
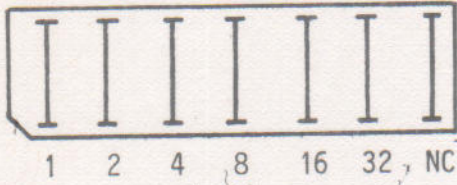


FIGURE 19. WRAPPING IN INTERRUPT

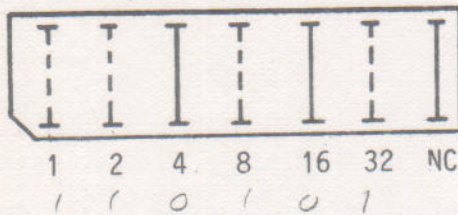
Code Plugs

Both interface cards and memory cards have code plugs to identify them. The code plug is binary coded, where the least significant bit will be found at the same side of the plug as the cut corner. Each connected wire-strap on the plug represents a binary "0", and each removed wire-strap represents a binary "1". The plug shall be mounted in the same direction as the IC on the card.



Logical Select Code

Each interface card has its own logical select code plug, identifies the interface card towards the program. For instance; a select plug with code 53_8 will look like this:

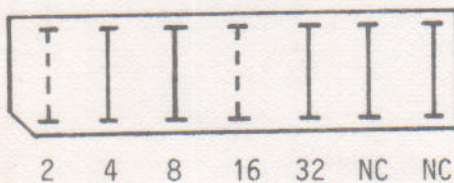


75

101 111

Logical Address Code

Each memory card has its own logical address code plug, giving the memory card its first logical address. The plug is coded module 2 KB, and each bit on the plug has an absolute value. On memory cards with a size larger than 2 KB, all bits lower than the memory card size are invalid. For instance; a 2 KB memory card with its first address at 44000_8 will have the following code plug:



Note. Some memory cards differ from this standard coding. Confirm with the data sheet of the particular card for the coding performed.

Figure 20. CODE PLUGS