STANSAAB

Alfaskop 3500 Data Terminal System

SAS

Technical Description

EE563–412
EE557–412
EE559–412
EE560–412

1. General Concepts

2. Control Unit 3501

3. Display Unit 3510

4. Display Station 3550

STANSAAB

Introduction

Principles of Operation

Software Information

Alfaskop 3500 Data Terminal System

Technical Description

General Concepts

EE563–412
1975–01–10

Introduction

Contents

GENERAL 1

DESIGN 2

- Control Unit 2
- Display Unit 3
- Display Station 3
- Data Printer 3

TERMINAL SYSTEM CONFIGURATION 4

SYSTEM COMPONENTS 5

FIGURES

1. Display Station 3550 / Display Unit 3510 2
2. Data Printer 3571/3572 3
3. Alfaskop 3500, remote application 4
4. Alfaskop 3500, local application 4
5. Control Unit, block diagram 5

GENERAL

The Alfaskop 3500 Data Terminal System features a new generation of system components that can be combined to suit different requirements regarding data terminal layout. The information intended for service personnel is contained in the following documents

- Technical description
- Application description
- Installation and maintenance manual
- Spare parts list
- Modification description

The technical description, the installation and maintenance manual and the spare parts list provide general information about the data terminal system. Information about a certain project is contained in the application description which also might include technical description of special units.

This technical description is divided into the following parts

- General concepts of Alfaskop 3500
- Control Unit 3501
- Control Unit 3502
- Display Unit 3510
- Display Station 3550
- Data Printer 3571/3572
- Special options

The following parts contain the hardware descriptions of the units. The rest of "Introduction" of this part contains information to help readers acquire a general understanding of the interrelationships among the system component of Alfaskop 3500. Thereafter follows a part "Principles of operation" where the main functions of the control logic are discussed, and a part "Software information" where the "interface" between hardware and software is discussed.

The drawings and diagrams are inserted at the end of this book. On some drawings there are Swedish text that contains information for the factory only. Please disregard all Swedish text that is not translated.

DESIGN

The Alfaskop 3500 Data Terminal System includes four basic components

- Control unit (for cluster configuration)
- Display unit (for cluster configuration)
- Display station (stand-alone)
- Data printer

In addition to the basic components, the Alfaskop 3500 Data Terminal System offers a number of options.



Fig. 1. Display Station 3550 / Display Unit 3510

Control Unit

The control unit enables the Alfaskop 3500 to be connected to a data processing system. There are two basic versions

- Control Unit 3501 – designed for connection to data processing systems via modems (remote applications)
- Control Unit 3502 – designed for connection to IBM System/360 or System/370 data channels (local applications)

Data Printer

The data printer provides hard copies of information appearing on the display screen or data obtained from the data processing system. There are two basic versions, both designed for connection to Control Unit 3501 or Display Station 3550.

- Data Printer 3571 – speed 30 ch/s
- Data Printer 3572 – speed 120 ch/s

Display Unit

Display Unit 3510 is used in cluster configurations to present data transmitted from the data processing system. It is designed for connection to Control Unit 3501. The optional Keyboard 3540 is connected to the display unit via a 1-metre cable and is thus movable.

Display Station

Stand-alone Display Station 3550 is designed for connection to a data processing system via modems (remote applications). It incorporates the requisite control electronics and a modem interface. The optional Keyboard 3540 is connected to the display station via a 1-metre cable and is thus movable.

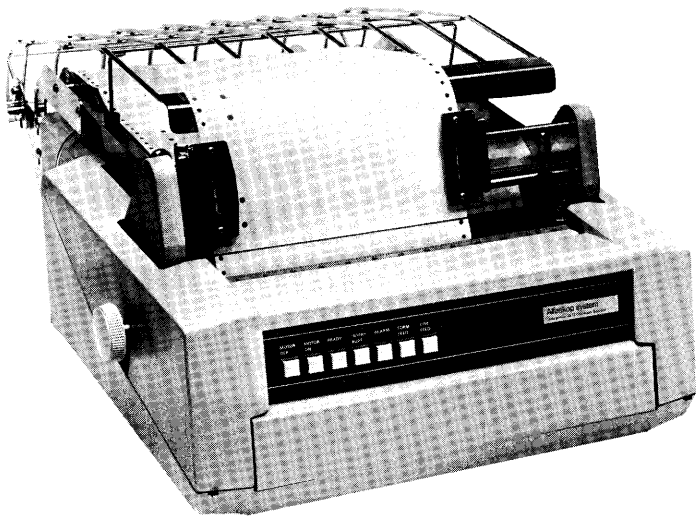


Fig. 2. Data Printer 3571/3572

TERMINAL SYSTEM CONFIGURATION

The Alfaskop 3500 Data Terminal System is ideal for a wide variety of applications including stand-alone, cluster, remote and local terminal systems. In remote applications, the Alfaskop 3500 includes Control Unit 3501 or Display Station 3550.

In local applications, the Alfaskop 3500 includes Control Unit 3502 which is connected to IBM System/360 or System/370 via a multiplexor or selector channel.

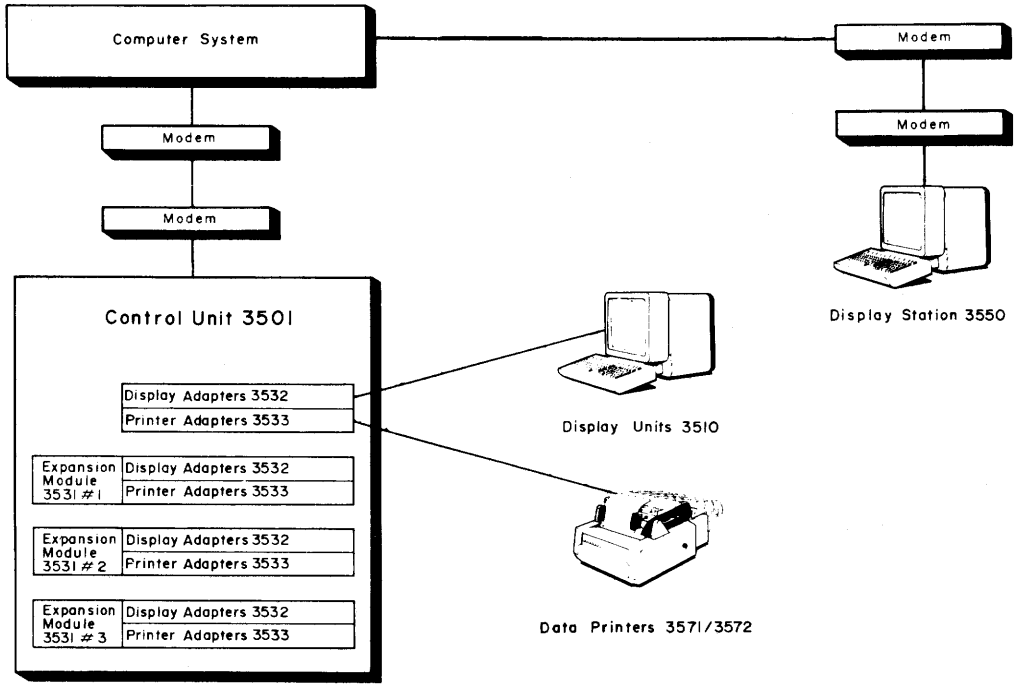


Fig. 3. Alfaskop 3500, remote application

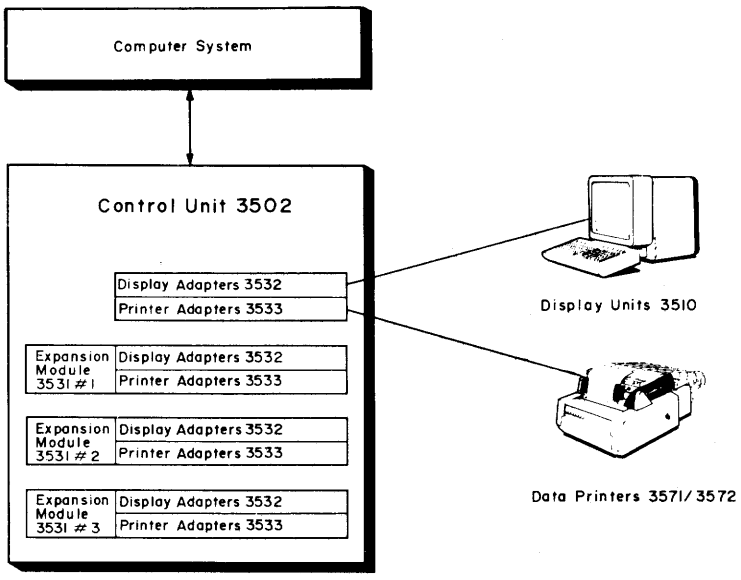


Fig. 4. Alfaskop 3500, local application

SYSTEM COMPONENTS

The Alfaskop 3500 Data Terminal System is highly flexible. In addition to technical and operational features, this flexibility embraces the design and construction of the hardware. Hardware components are designed for both stand-alone operation and cluster configurations. Modular design philosophy enables the Alfaskop 3500 to offer the user a wide range of configurational options.

The following Alfaskop 3500 components are intended for *stand-alone applications*

- Display Station 3550 – designed for modem connection. This display station consists of a CRT unit which incorporates the requisite control logic. Optional components include
  - Keyboard 3540
  - Identifier 3541
  - Light Pen 3542
  - Printer Adapter 3561, used to connect Data Printer 3571 or 3572

Cluster configurations are used in installations where more than one display unit or more than one data printer are to be connected to the same communication channel. This means that all electronics that control communication, input/output, presentation etc. will be assembled within the control unit, to which the display units and data printers are connected via adapters. The display units and data printers can be located up to 600 metres from the control unit.

The following components are used for cluster configurations

- Control Unit 3501 – designed for modem connection. The basic version of Control Unit 3501 includes a communication adapter for modem transmission as well as the control logic needed to control eight display units and eight data printers via adapters. If so desired, the capacity of Control Unit 3501 can be easily expanded by adding expansion modules or control moduled. A maximum of three modules can be installed in each Control Unit 3501, thus providing a total capacity of 32 display units and 32 data printers. Optional components that can be mounted in Control Unit 3501 include

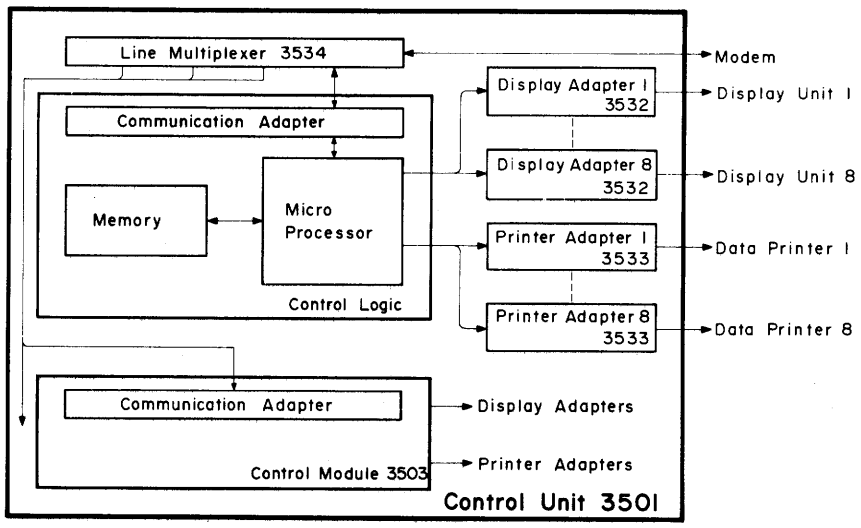


Fig. 5. Control Unit, block diagram

- Expansion Module 3531, contains the control logic needed to connect eight additional display adapters and eight additional printers to Control Unit 3501
- Display Adapter 3532, used to connect Display Unit 3510
- Printer Adapter 3533, used to connect Data Printer 3571 or 3572
- Control Module 3503, is designed in the same way as the control logic of the Control Unit 3501 and can thus serve up to eight display adapters and eight printer adapters. The control module functions independently of the basic circuits of Control Unit 3501
- Line Multiplexer 3534 is used when there is more than one communication adapter connected to the same modem equipment

- Control Unit 3502 – designed for direct connection to an IBM data channel. The basic version includes a communication adapter for channel transmission as well as the control logic needed to connect up to eight display units and eight data printers via adapters. Here too, capacity can be increased using expansion modules. Control Unit 3502 permits the connection of three expansion modules, thus providing a total capacity of 32 display units and 26 data printers. The following optional components can be used with Control Unit 3502. See previous paragraph.
  - Expansion Module 3531
  - Display Adapter 3532
  - Printer Adapter 3533

- Display Unit 3510 – designed for cluster configurations. The following optional components can be included
  - Keyboard 3540
  - Identifier 3541
  - Light Pen 3542

Principles of Operation

Contents

GENERAL 1

DISPLAY STATION 3550 1

- Memory 1
- Micro Processing Unit 2
- Modem Communication Adapter 3
  - Receiving 3
  - Sending 4
  - Monitoring 4
- Printer Adapter 3561 4
- Display Logic 5
- Display and Keyboard Adapter 5
- Optional units 6

CLUSTER CONTROL UNITS 6

- Memory distribution and addressing 6
- Timer 8
- Multiplexor 8
- Display Adapter 8
- Printer Adapter 3533 9
- External Bus Adapter 9

FIGURES

1. Block diagram of Display Station 3550
2. Block diagram of Control Unit 3501 7

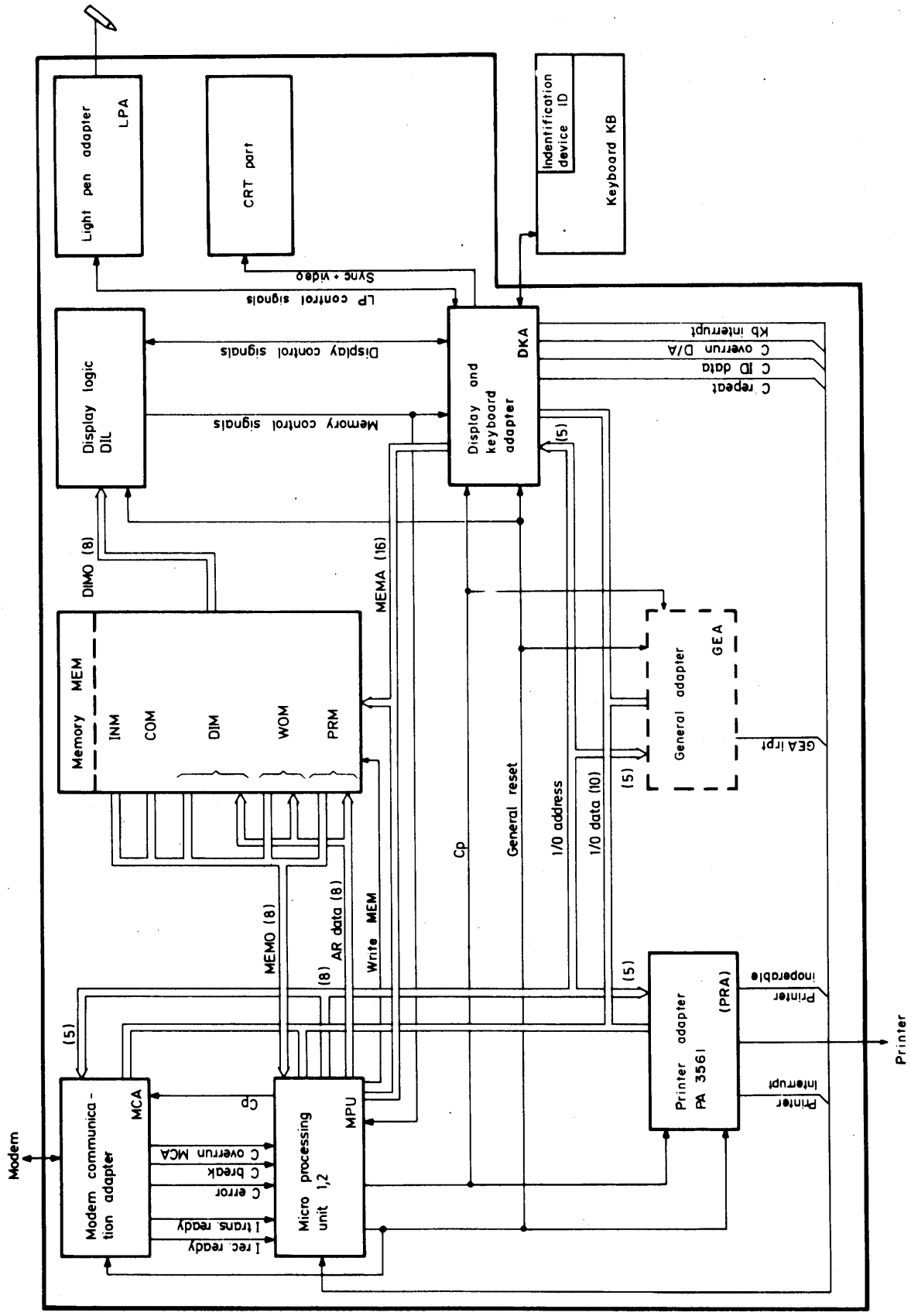


Fig. 1. Block diagram of Display Station 3550

GENERAL

All functions in the control logic are controlled by a micro processing unit (MPU) which performs operations according to the programs that are stored in a memory. The only exception is the presentation function that is self controlled.

The adapter units, such as modem communication adapter, display and keyboard adapter etc., can be regarded as peripheral units to the MPU. They can send condition signals to the MPU and they can request service by sending interrupt signals to the MPU. Data is exchanged via an I/O data bus either from a peripheral unit to MPU or from MPU to a peripheral unit. The MPU controls the exchange by issuing control and address signals on an I/O address bus. There are no connection whatsoever between the peripheral units.

These general principles apply to both single and cluster configurations althrough the latter are somewhat more complicated. In the following the block diagram of the single station (Display Station 3550) is first discussed and then follows a discussion of the cluster block diagram.

The blocks of the block diagrams in Fig. 1 and Fig. 2 correspond to one PC-board (printed circuit board). The MPU comprises two PC-boards and the memory can comprise one or two PC-boards depending on program complexity.

DISPLAY STATION 3550

The block diagram of the display station is shown in Fig. 1. The two blocks that are dashed on the diagram can be inserted in a PC board position that is reserved for special options. Two small PC boards (half board) or one full size PC board can be inserted. The printer adapter is only included if a printer is connected.

Memory

The memory can comprise one or two PC-boards of different types depending on needed capacity. Functionally it can be divided into the following parts

- INM instruction memory (ROM)
- COM constant memory (ROM and constant bytes)
- DIM display memory (RWM)
- WOM working memory (RWM)

These four parts are always included. If a printer is connected

- PRM printer memory (RWM)

also is included.

The instruction memory contains all the instructions that build up the programs. The programs specify the operations to be carried out within the display station and consequently they are different in different applications. The ROM is expandable in steps of 256 bytes. It is easy to exchange ROMs and alter the operation.

The constant memory contains information that is needed when the program is running. This information can be of two types either related to an application or individual for different work stations. Application dependent information (e.g. display screen format) is usually stored in ROM and individual information (e.g. address of the terminal) is usually stored in constant bytes.

The information in the instruction memory and the constant memory can be used only by the MPU. The MPU addresses a memory cell via MEMA (memory address bus) whereupon the byte contained in this cell is fed to the output of the memory (MEMO).

The information that is presented on the display screen is stored in the display memory. There is one memory cell for each position on the screen. The display memory is the only part of the memory that can be addressed from another source than the MPU. When a line of the display screen is to be presented, the display logic (DIL) needs information about all characters on this line. During the reading of these characters, the memory is addressed from the display and keyboard adapter.

All writing on the screen (that is to say in the DIM) is made by the MPU and is thus controlled by the program. The MPU puts the byte to be written on the AR data, addresses the memory (MEMA) and sets the Write MEM signal.

The working memory is used to store data temporarily while an operation is performed. Different parts are used for different purposes e.g. I/O area for storing information to or from the modem, keyboard area for storing information from the keyboard.

The printer memory is used as a buffer area for the printer. The information can come either from the modem or from the display memory. If the content of the screen is to be printed out the information is transferred from DIM to PRM. After that, the DIM is free to receive new information while the printer is operating.

Micro Processing Unit

The micro processing unit controls all data exchange between the MPU and the peripheral units, all writing into the memory and all data processing (except presentation).

The MPU work is determined by instructions received from the memory. Each instruction gives rise to several events according to microinstructions stored within the MPU. In order to perform the operations specified by an instruction, several microinstructions must be executed; the number of microinstructions depends on the complexity of the instruction at hand. A program that controls an operation of the data terminal (e.g. insert line) embraces a sequence of instructions.

Interrupt signals are received from the peripheral units when they have performed an operation such as sent or received a character. Since the MPU actions started by an interrupt signal are of different importance and need to be carried out within different times, the interrupts are arranged in priority order. If no interrupt signal has arrived, the MPU works with a background program which has the lowest priority level. When an interrupt signal has arrived, interrupt condition is generated. After

the instruction being executed when the interrupt signal arrived is fully executed, the information needed to restart the program is stored. Then, a program that handles the interrupt signal is started.

The actions of the programs started by the interrupt signals are organized in two main procedures. One procedure is to carry out all operations on the interrupt level and thereafter return to the background level. The other procedure is to set a flag bit in a register and immediately return to the background level. In such case, the operations are carried out on the background level when the program finds the flag bit. Of course, several programs follows procedures somewhere between these extremes.

When a program is running the MPU feeds the address of the instruction after the one being executed to the memory. The address of the next position on the screen in connection with data entry is also stored in the MPU. Since the address bus is common to the presentation and the MPU, one of the sources must have priority over the other. The presentation cannot wait when once started and has thus the highest priority. That means that the MPU work must be stopped while data is transferred from the memory to the display logic.

The instruction set of the MPU is adapted to the needs of a data terminal. Thus, the instruction set does not embrace some instructions that are common in conventional mini-computers.

Modem Communication Adapter

The modem communication adapter (MCA) is used to adapt modem-communication signals in accordance with the requirements imposed by the CCITT V24 interface. It is provided with a strapping plug which enables the transmission rate, transmission type, transmission code etc. to be selected. The transmission procedures are determined by the MPU and programs. Sending and receiving of characters as well as synchronization is carried out within the MCA.

Receiving

In a remote terminal system, all terminals must be in the receiving state from the outset. It is always the computer that initiates communication.

The modem communication adapter receives signals from the modem. In asynchronous transmission the MCA issues an interrupt signal for each character received. In synchronous transmission the MCA must detect two consecutive SYN characters before issuing interrupt signals.

When the MPU handles the interrupt signal it sends signals across the I/O address bus. These signals are decoded as instruction read receiver by the MCA which gates out the received character onto the I/O data bus. The MPU reads the I/O data bus after an ample time and takes away the signals on the I/O address bus.

If the MPU has not fetched the previous character when a character arrives, the MCA sends the condition signals C error and C overrun MCA when the interrupt signal is sent. These condition signals are not reset until another character has arrived and the next interrupt signal is issued. If the MCA detects a parity error, the condition signal C error is sent. C break is sent if the stop bit is missing in connection with asynchronous transmission.

Sending

The sending circuits of the modem communication adapter are controlled by the MPU via signals across the I/O address bus. These signals are decoded to Set request to send, Reset request to send and Write transmitter.

The MPU sets the MCA in send-mode by ordering Set request to send. Then the MCA immediately sends the signal request to send to modem. When the modem is ready it sends the signal ready for sending to MCA which issues interrupt signal. The MPU then feeds the characters to be sent (in synchronous transmission the first characters must be SYN) to the I/O data bus and starts the transmission by sending Write transmitter via I/O address bus.

Each time the MCA has completed sending of a character to the modem, it issues an interrupt signal to the MPU. When the MPU has sent all characters belonging to a message, it sends Reset request to send via I/O address bus. After the MCA has completed the sending of the last character it returns to the receive state.

Monitoring

The display station is provided with a lamp (SYSTEM READY) which

- is extinguished when the line is inactive
- flashes when the modem is only receiving data
- glows steadily when the control unit is sending data

The monitoring logic is a part of modem communication adapter.

Printer Adapter 3561

The printer adapter 3561 (PRA) receives characters in parallel form from the I/O data bus and transmits them to the printer in serial form (asynchronous transmission, asynchronous character structure). It receives a signal indicating whether the printer is operable or not from the printer and this signal is sent to the MPU as a condition signal. In some applications a signal (Reversed channel) from the printer can be used within the PRA to stop transmission.

In the printer adapter only one instruction, data to printer adapter, can be decoded from the information on the I/O address bus. However, the most significant bit of the character on the I/O data bus indicates if it is intended as an order character for the PRA or a character to be sent to the printer. The order character can call for starting of the printer motor (which is accomplished by PRA sending Request to send) followed by an interrupt signal after ample time. It can also call for stopping of the printer motor after a suitable delay time. The time is specified by the order character.

The characters transferred to the printer comprise both control characters and text characters according to a code table. Each time a character is sent from PRA, PRA issues an interrupt signal to the MPU which sends the next character etc. The printer is supposed to handle the characters at the speed they are transmitted. If a control character, which calls for a time-consuming action within the printer is sent it should be followed by an order character causing a delay that enables the printer to finish its action before next character arrives.

Display logic

The display logic (DIL) generates control signals and video signal for the CRT part. One part of the memory, the display memory (DIM) is used to store the information to be presented on the screen. Each position on the screen corresponds to one 8-bit memory cells.

The display logic operates self-governed. 50 times a second the information in the DIM is presented on the screen. DIL generates a vertical sync pulse which indicates that the presentation of the next frame is to begin and a horizontal sync pulse which indicates the start of a new sweep. Internally, it uses approx. 100 position signals for each sweep. These position signals control the video generation. Several position signals are also fed to the display and keyboard adapter which uses them as timing signals.

Before a text line is presented, DIL fetches the information for the complete line from the memory. In the mean time, the beam makes two idle sweeps (extinguished). When characters are generated the information for one line is scanned during nine sweeps. Thereby the video signal is composed in such a way that the beam will make light dots in a pattern that generates the symbols.

The video signal as well as the sync signals are fed to the display and keyboard adapter which sends these signals to the CRT part.

Among the characters in the DIM are different control characters used to define different fields on the screen e.g. the brightness level for the following characters. These characters, which occupy one position each in the DIM, are decoded in the display logic. For line end or frame end no special characters are needed since the DIL logic itself controls the presentation format.

Since DIL and MPU address the display memory via the same memory address bus, DIL sends a signal to MPU before fetching information. This signal is sent in advance to allow the MPU to stop its work at a convenient point of time. Then DIL sends a signal which allows addressing only from the display and keyboard adapter (there is no space for the memory address counter on the DIL PC-board, therefore it is located on the display and keyboard adapter). The memory address counter is stepped by signals from DIL while character after character for the line at hand are fetched.

Display and Keyboard Adapter

The display and keyboard adapter (DKA) comprises the interface between the control logic on one hand and the CRT part, keyboard and the optional units identification device and light pen adapter on the other hand. All data transfers are made in parallel form. It also holds the memory address counter used when a text line is transferred from DIM to DIL.

The correct timing of the sync pulses and video is controlled by the display logic. After the start of the V-hold signal (vertical sync) sent to the CRT, there are several idle sweep before the presentation of the first line starts. During this time the display and keyboard adapter transfers data to the MPU.

The MPU can send commands to CRT part, ID logic or light pen logic at any time. It feeds the command byte onto I/O data bus and enters the command into DKA by signals on I/O address bus. Some of the command bits should be active only a time corresponding to one frame. These bits are partly controlled by DKA that uses the V-hold for timing. The other command bits are timed by the MPU setting and resetting them.

Data to the MPU can originate in the keyboard, the identification device or in the light pen adapter, and be of different type. The DKA is provided with a priority logic that determines which of the data sources to read. Then the DKA sends a condition signal according to the selected information and issues the KB interrupt signal. The MPU fetches the data by signals on the I/O address bus which are used in the DKA to gate out the byte onto I/O data bus. The DKA uses position signals from the display logic for timing of the priority logic, reading and interrupt signal.

Only one interrupt is issued each frame. Thereby the MPU has time enough to fetch the data. If, however, the MPU does not fetch the data byte before the next byte arrives, the condition signal C overrun DIA is sent to MPU.

Optional units

The keyboard is included in most applications. An identification device can be mounted in the keyboard when only authorized operators should have access to the computer.

There is one PC-board position that can be used for optional adapters. This position is provided with connections to the display and keyboard adapter (intended for light pen adapter signals) and to the I/O address and I/O data buses.

The light pen can be used to select among light pen detectable fields on the screen. Data from the light pen contains information about which field the operator has selected.

The connections to the I/O buses can be used by an adapter designed to interface I/O units such as cartridge recorders, cassette tape units, paper tape punches etc.

CLUSTER CONTROL UNITS

The principles of operation are the same in cluster and single variants. In the following description the details that differs are discussed. In installations where more than one display unit is needed, the capacity of the MPU is more exploited if it controls several display units.

Fig. 2 shows one PC-board mounting frame of a cluster control unit. Each frame accomodates all logic needed to control up to eight display units and up to eight printers. The two MPU PC-boards, one or two memory PC-boards, one multiplexor PC-board (MUX) and one timer PC-board (TIM) is always inserted in a control unit. For each display unit that is connected one display adapter is inserted and for each printer one printer adapter is inserted. For data exchange with the computer a modem communication adapter (remote operation) or a channel communication adapter (local operation) is inserted. For data exchange between MPUs an external bus adapter is used. The configurations for data exchange are further explained in the parts describing Control Unit 3501 and Control Unit 3502.

Memory distribution and addressing

The display memory is distributed so that the display areas for different display units are mounted

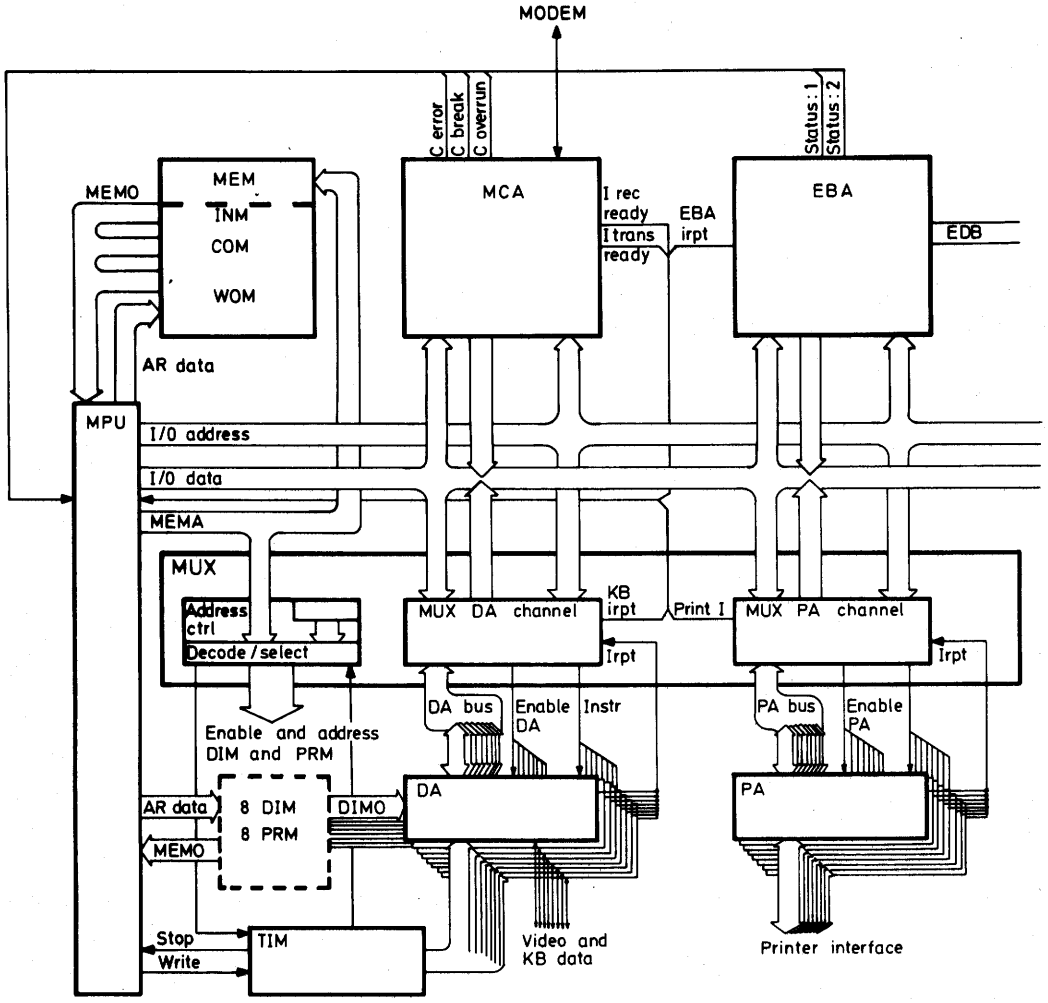


Fig. 2. Block diagram of Control Unit 3501

on the display adapters. The printer areas are mounted on the printer adapters. Thus the memory capacity is increased only when needed. On the memory PC-boards the instruction, constant and working memory are mounted. Some minor parts of the working memory, that are related to the individual display units, are contained in the display memories.

The MUX controls the addressing of display memory (DIM) and printer memory (PRM). It contains a memory address counter that is used when the information for a text line is transferred from DIM to the character generation circuits on the display adapter. This transfer takes place simultaneously on all display adapters. The MPU can thus address any cell on the memory PC-boards without any risk of mixing addresses.

When the MPU addresses a cell of DIM or PRM the address control circuits on the MUX feed the 11 least significant bits of MEMA to all display and printer adapters and one enable signal to the adapter at hand (decoded from the 5 most significant bits of MEMA). This addressing is impossible when information is transferred from DIM to the presentation circuits. Therefore the address decoder

in MUX feeds a signal to the TIM if a DIM or PRM address is decoded. If the TIM at that instant issues timing signals for data transfer from DIM, it sends a stop signal to MPU whose work is stopped immediately.

Timer

The timer (TIM) contains the timing circuits for the display adapters. The presentation is carried out simultaneously in all display adapters, i.e. V-sync, H-sync etc. are generated in the TIM and feed to all DAs. Also timing signals for data transfers between display adapters and display units are generated in the TIM.

Multiplexor

In a cluster configuration, the MPU has no possibilities to receive interrupt and condition signals from all adapters. The MUX has one DA channel and one PA channel that multiplexes these signals so that at every moment the MPU senses only one display or one printer adapter.

In the DA channel there is a scanning counter that enables the eight display adapters one after another. If an activated interrupt signal is gated out from an adapter, the scanning counter is stopped, whereupon the MUX issues KB interrupt signal to the MPU.

The MPU then fetches two bytes from the MUX, each by sending signals via the I/O address bus that cause MUX to gate out the byte onto I/O data bus. The first byte contains information about the value in the scanning counter i.e. the address of interrupting DA together with status signals which replace condition signals from that DA. The second byte is the data byte from keyboard, identification device or light pen. After the second byte is fetched, the scanning proceeds.

If the MPU is to send a command to the display adapter, it transfers two bytes on the I/O data bus. Each of the bytes is transferred controlled by signals on I/O address bus. The first byte contains an address and is set in a register in the MUX, which selects the addressed DA and stops scanning. The second byte is transferred to the selected DA, whereupon the scanning continues again.

The PA channel works in the same way, but since no data is transferred from the printer only one byte, the address and status byte, is fetched by the MPU after interrupt.

Display Adapter

The display adapter (DA) contains a display memory (DIM), character generation circuits and logic for serial transfer of video and data bytes via a coaxial cable which connects the display unit. The timing of the DA circuits is controlled by the TIM which means that all eight DAs work in parallel. The character generation circuits (together with TIM) work in the same way as in the display logic (DIL).

To control the display screen, a vertical sync pulse is transferred at the beginning of each frame and a horizontal sync pulse is transferred at the beginning of each sweep. During 24 sweeps after the vertical sync pulse no video is transferred. These 24 sweeps are used to transfer one data bit per sweep. One command byte is transferred to the display unit and one tag byte and one data byte are transferred to the display adapter. The transfer principles are described in the 3501 part.

Printer Adapter 3533

The Printer Adapter 3533 that is used in cluster control units operates in principle as the Printer Adapter 3561 described in the previous section "Display station 3550". Data exchange with the MPU is carried out via the MUX. One difference is that PA 3533 can receive a signal indicating parity error from the printer. This signal is sent to the MUX at proper time with respect to character transmission. The MUX includes this signal as a bit in the status byte sent to the MPU after interrupt.

External Bus Adapter

The external bus adapter (EBA) is used for data transfer between micro processing units across an external data bus (EDB). Thus it is only inserted if the control unit comprises more than one frame. Each frame requires an EBA. The EDB comprises 15 lines of which 9 lines carry bits in a data character and 6 lines carry control and status signals. The data exchange is program controlled. In the programs the frame that holds a communication adapter is defined as master and the other frames are defined as slaves. In addition switches on the frames must be set in position master and slave respectively.

The master unit controls all transmission across the EDB. A status register with four bits can be used to check the transmission procedure. The MPU can enter information into the status register by putting data on the I/O data bus and sending signals on the I/O address bus. One of the status bits (bit 0) is used internally in the EBA (not sent on EDB). This bit indicates if the EBA in question is participating in transfer or not. One of the status bits can be used only in the master; it initiates general reset in all slaves. Two other status bits are transferred on the EDB.

Connection between two frames is established by the master (or another unit that has got permission from the master) sending general call using one control line. Then status bit 0 is set in all EBAs but it will remain set only in those who are interested to participate in communication. All units have to check the character that is transferred via the data lines in connection with general call to determine their interest. Data is then transferred across the data lines between EBAs having status bit 0 set controlled by a "handshake" procedure on two of the control lines.

The MPU in a sending frame puts the character onto the I/O data bus and sends signals on the I/O address bus. The EBA immediately feeds the character onto EDB, and when all receiving units have taken care of the character, EBA issues an interrupt signal to MPU etc. In a receiving EBA, an interrupt to MPU is generated each time a valid character is present on EDB. Then the MPU fetches the character via I/O data bus by issuing signals on I/O address bus.

Software Information

Contents

GENERAL 1

INSTRUCTIONS 1

- Micro Processing Unit 1
- Instruction formats 2
- Instruction list 3

INTERRUPT SIGNAL SYSTEM 7

FIGURE

1. Simplified block diagram of the MPU

APPENDIX

1. Instruction summary

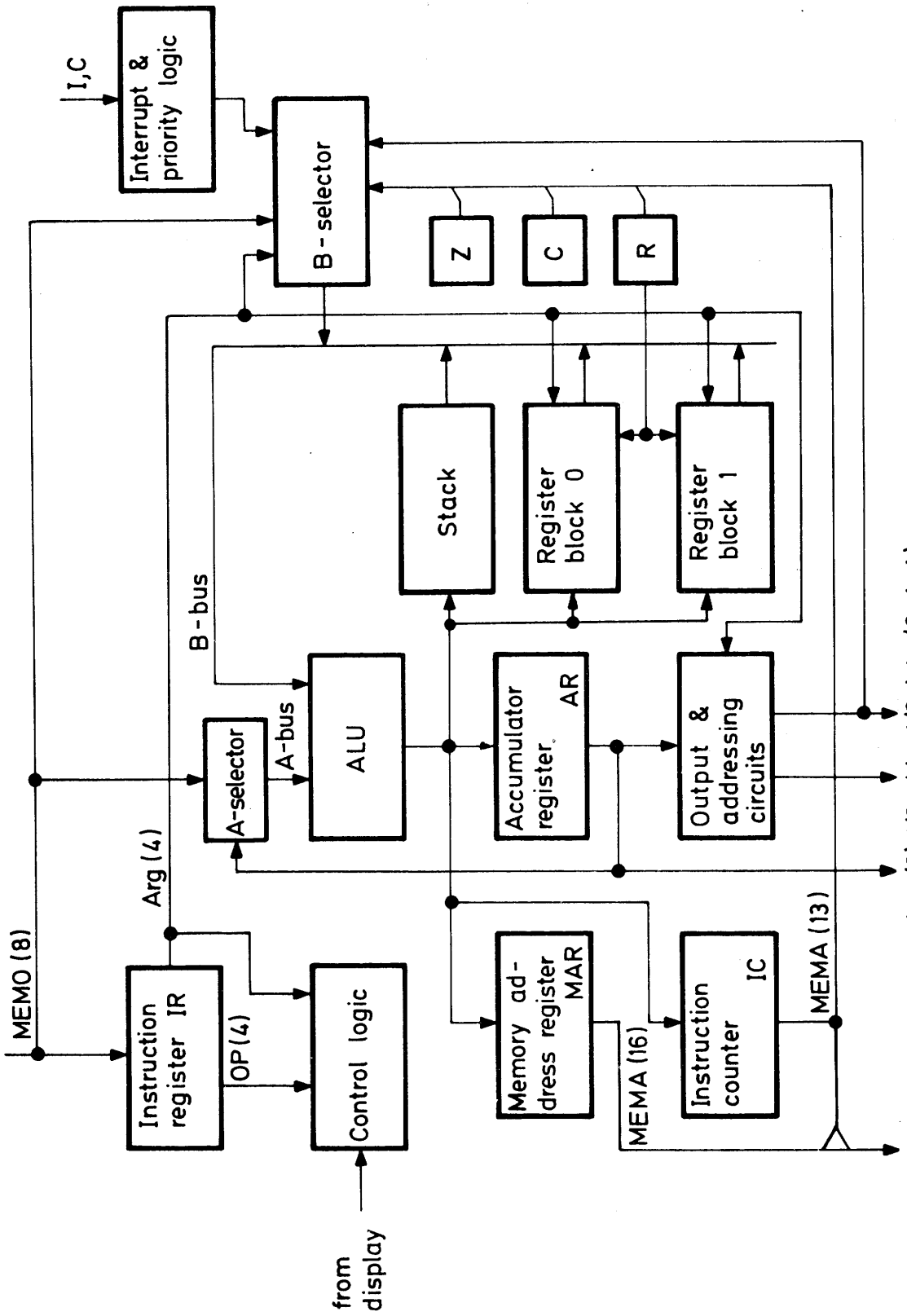


Fig. 1. Simplified block diagram of the MPU

GENERAL

The operation of the terminal is to a great extent determined by the programs stored in the instruction memory. Therefore the hardware description must be supported by a program description to fully explain the operation. In the following some ways of describing software is discussed. First, however, the instruction set is presented and then follows a description of how some hardware facilities are used in the programs.

INSTRUCTIONS

Micro Processing Unit

The most important parts of the micro processing unit (MPU) are shown in the block diagram in Fig. 1. If the reader is familiar with this block diagram, he has easier to understand the instructions and programs. Some of the blocks (the selectors, control logic and the ALU, arithmetic logic unit) are not mentioned in the explanation of the instructions, but are nevertheless essential for the comprehension of MPU operation. The number of bits on the data buses is written within parentheses.

- The instruction register, IR, is an eight bit register that holds the instruction (or one byte of a multiple byte instruction) being executed. The four most significant bits of the instruction register are called the OP-part (operation part) and the four least significant bits are called the argument (Arg). In most cases the OP-part defines what is to be done, and the argument is used as address bits (to registers, input/output units and other circuits).

- The control logic gets information about current instruction and converts this information into control signals sent to the rest of the MPU. The control signals make sure that the events to be carried out during execution (fetching of instructions, transfer of data, setting of registers, checking of conditions etc.) are carried out in correct sequence and at proper time. The signals informing that the display logic is going to use the memory buses are also fed to the control logic.

- The interrupt and priority logic stored the interrupt signals issued by other units. It also holds the priority of current program. If the priority of current program is lower than the priority of the interrupt signal, the current program is interrupted. The start address for the interrupt program is formed in the interrupt and priority logic.

- Two 8-bit data buses are fed into the ALU. The information on the buses is selected by the A-selector and B-selector. The ALU can perform logic and arithmetic operations on the bytes present at the A- and B-inputs. All data transfers within the MPU proceed via the ALU. During transfers the ALU simply moves data from one input to the output.

- The accumulator register, AR, is the working register of the ALU. It often holds an intermediate result when the MPU proceeds from one instruction to the next. Information that is prepared for storing in the memory or for writing into the other units (e.g. modem communication adapter) is also put in the AR. Then the MPU sends out control signals (Write MEM to the memory,

Write transmitter via the I/O address bus to the modem communication adapter etc.).

- The memory address register, MAR, is a 16-bit register that is capable of addressing any memory position, i.e. up to 64K. It is used in connection with reading and writing in the RWM.

- The instruction counter, IC, contains the address to the next instruction (or to the next byte in multiple byte instructions). IC contains 13 bits and is thus capable of addressing up to 8192 (8K) bytes. The 12 least significant bits are contained in an ordinary counter. The 13th bit (also called extension bit) can be changed by the program as well as by normal counting.

- The stack is used for temporary storage of data. Only the last stored byte is available at the output, but when it is fetched the next last byte is available etc. By way of example, at interrupt the current data in the IC, R, C, Z and AR are stored to be retained when the interrupted program is started again. Up to a maximum of 48 bytes can be stored in the stack.

- The two register blocks each contain 16 addressable 8-bit cells. A register block selector, R, that can be actuated by the program, selects one of the blocks. The argument of the current instruction is used to select one of the 16 cells within the block. The registers are used in connection with, among other things, arithmetic and logic operation and memory access.

- There are two condition flip-flops (carry, C, and zero, Z) that can be set during execution of the instructions. They are used as mark bits for the program.

- The output and addressing circuits gate AR data onto the I/O data bus (and generate the parity bit). Addressing of the other units in connection with input and output is also carried out in this block.
  (Input data from the I/O data bus is fed to the B-selector.)

Instruction formats

The instructions can contain operation part (OP-part or OP), memory address, condition, register address and data. OP-part is always included and it can comprise four or eight bits (or, in some multiple byte instructions, even twelve or sixteen bits). Other parts are included as needed.

The memory addresses can comprise either a 12-bit absolute address or an 8-bit relative address. A relative address comprising eight bits states a number (−128 to +127) which will be added to the current number in the instruction counter. Conditions and register addresses comprise four bits. Data comprises eight bits.

The instructions comprise one, two, three or four bytes. They have different formats as shown in Appendix 1, table 4. Functionally, they can be divided into groups as: input/output instructions, program affecting instructions, internal data transfer instructions, memory reference instructions and logic instructions. Each instruction can be denoted either by a hexadecimal number (machine code) or according to a mnemonic code.

The contents of a byte (register, bus, stack, memory cell etc.) are denoted bit 0 to 7 where bit 0 is the least significant bit. Any parity bits is denoted PC (parity check) and is considered as the most significant bit.

During execution, the instructions are fetched from the instruction memory on the memory PC-board. Every time a byte is entered into the instruction register, the instruction counter is incremented by one. Thus the next byte or instruction in general has arrived from the memory when the first instruction (or byte) is finished.

SWI

If a memory cell is not programmed, its contents are interpreted as FF (which means that no operations are carried out until a programmed cell is encountered). If the memory address refers to a cell within a block that is not mounted on the memory PC-board, 00 is entered into IR.

From Appendix 1, table 4, it is evident that only a few instruction formats seem to suit the IR division into OP and arg. However, the sequence of microinstructions that the control logic uses are adapted to execute all instructions efficiently. The four most significant bits of the first byte (OP-part) are first considered when a certain sequence is selected. Then additional operational information is considered as needed (to allow the MPU to operate on the last OP-part in the second byte, OP and arg can change place in the IR).

Instruction List

The tables in Appendix 1 present the instructions in abbreviated form. Therefore only some additional information on most instructions is given below. It is evident from table 1 that the four most significant bits of the first byte determine, in most cases, the type of instructions. In the following description the hexadecimal notation of these four bits is used for classification. The conditions of the relative jumps shown in table 1 are only examples. The input and output instructions may be changed in some applications.

Program affecting instructions

0, JUMP
1, CALL

There are four jump instructions (0–3). The condition flip-flops (C and Z) are not affected when any jump instruction is executed. The jump instructions are used when the program continues at another address in the instruction memory. If the instruction memory comprises more than 4K bytes, 13 bits must be used when addressing. The 13th bit (extension bit) must be handled separately when instructions 0 and 1 (JUMP and CALL, format 4) are executed since the address given in the instruction comprises only 12 bits. When an absolute jump is made, the address given in the instruction is entered into the instruction counter.

4B, EXTN

If the jump calls for an address outside the current 4K area, instruction 4B (EXTN) must be given before instruction 0 or 1. Then the extension bit is changed before the jump is carried out. Interrupts are inhibited until the jump is made. Note that the extension bit can be changed also in the normal way when instruction counter is incremented or decremented.

1, CALL
4F, RTRN

Before the jump in a CALL instruction is made, the address of the second byte of the instruction is stored in the stack (together with R, C and Z). Instruction 4F (RTRN, format 1) can be used to go back. It fetches the address (and R, C, Z) from the stack, whereafter it causes a jump to this address +1, which means the address to the instruction after the CALL instruction.

2, JCTc
3, JCFc

Instructions 2 and 3 (JCTc and JCFc, format 5) cause a jump only if the condition specified in the instruction is true or false respectively. This is the definition of a conditional jump. The eight address bits specify a relative address. If the most significant bit is a ONE the seven other bits are interpreted as a negative number (in TWO's complement) otherwise they are interpreted as a positive number. The number of the relative address is added to the current value of the instruction counter. Thus the jump is made a certain number of steps relative to the former address, which is the definition of a relative jump.

Note that conditions 4 through E are adapted to the needs of different applications.

4A, ARIC

Instruction 4A (ARIC, format 1) causes a relative jump where the relative address is specified by the number in AR, (in the same way as for instructions 2 and 3). Note that this jump is unconditional.

4D, DITR
4C, ENTR

Instruction 4D (DITR, format 1) and 4C (ENTR, format 1) can be used where a sequence of instructions must not be interrupted. The instruction 4D overrides the priority conditions when interrupt signals are received.

4E, CPRM

When an interrupt signal causes an interrupt, the following information about the current program is stored in the stack:

- instruction counter (13 bits)
- conditions (Carry, C, and Zero, Z, – 2 bits)
- register block selector (1 bit)
- AR data (8 bits)

The priority of the interrupt is set and a jump is made to the start address of the interrupt program (extension bit is 0). The last instruction of the interrupt program should be 4E (CPRM, format 1). Then the information listed above is fetched from the stack, priority is reset and the execution of the interrupted program is resumed.

Logic and arithmetic instructions

8a, LDRa
9a,ANDa
Aa, IORa

Instructions 8–F have format 2, but if the argument is F (1111) they become two-byte instructions, format 3 (except FF that is a one-byte No op instruction). The condition flip-flops C and Z (carry and zero)

Ba, EXOa
Ca, ADDa
Da, SUBa
Ea, CMPa
Fa, STRa
FF, No OP

are set during execution of the instructions. They give the following information:

- Zero flip-flop
  - Instructions 8–D: zero is one-set if the contents of AR are zero after execution otherwise it is zero-set
  - Instruction E (compare): zero is one-set if the contents of AR are equal to the contents of addressed register (or second byte), otherwise it is zero-set. Note that AR is not changed during execution.
  - Instruction F (store): zero is not affected
- Carry flip-flop
  - Instructions 8 (load) and F (store): carry is not affected.
  - Instruction 9 (and), A (or) and B (exclusive or): carry is zero-set
  - Instruction C (add): carry is one-set if the result of the addition is greater than 255 (decimal number), otherwise it is zero-set
  - Instruction D (subtract): carry is one-set if the result of the subtraction is positive or zero. If the result is negative, carry is zero-set (and AR contains the number in two's complement)
  - Instruction E (compare): carry is one-set if the contents of AR are greater than or equal to contents of addressed register (or second byte), otherwise it is zero-set. Note that AR is not changed during execution.

42 a9, ADCYa
42 aB, SBCYa

There are two more arithmetic instructions, that have format 6 or format 10; add with carry (42 a9, ADCYa) and subtract with carry (42 aB, SBCYa). These instructions can be used when several bytes are to be added or subtracted. These instructions are carried out in almost the same way as instructions C and D respectively. The only difference is that the value of carry flip-flop is added.

Instructions for internal data transfer

The registers are used to stored frequently used data. Sometimes two registers are necessary to store data and therefore many instructions handle two register as a unit (even-odd register pair). The even-odd register pair is composed of two registers with consecutive addresses.

42 a0, SRINa
42 a1, SRDEa
42 p4, DRINp
42 p5, DRDEp

When the number in a register is incremented or decremented by one (42 a0, SRINa or 42 a1, SRDEa – format 6), zero is one-set if the number becomes 0 (otherwise zero-set) and carry is one-set if the operation gives carry (otherwise zero-set). (This means that, in connection with decrementing, carry is one-set in all cases except when the number becomes FF.) The same applies to register pair operations – incrementing or decrementing by one – (42 p4, DRINp or 42 p5, DRDEp – format 6), if the entire 16-bit pair is considered. The address p can

have a value of 0, 2, 4, 6, 8, A, C or E. If an odd value is given it is interpreted as the next lower value. Note that one byte in the stack is used during execution. AR is not affected.

42 a2 d, SRLDa
42 p6 d1 d2, DRLDp

Data is loaded to a single register by 42 a2 d, SRLDa, (format 9) where the third byte (d) contains data. The four byte instruction 42 p6 d1 d2, DRLDp (format 11) is used to load a register pair. AR, C and Z are not affected by these instructions, but one byte in the stack is used. The register pair address p must have a value of 0, 2, 4, 6, 8, A, C or E. The third byte d1 is loaded into the even register (eg 4) and the fourth byte d2 is loaded into the odd register (eg 5). If p is given an odd value, d2 will be loaded into the odd register while d1 will be lost.

43 ab, MORR ab
48 p1 p2, DRCM p1 p2

Two instructions have format 7:43 ab, MORR ab and 48 p1 p2, DRCM p1 p2. Instruction 43 transfers the contents of register b to a (a and b can have all values 0 through F). AR and C are not affected, but Z is one-set if the content (that came from register b) of register a becomes zero (otherwise it is zero-set). Instruction 48 calls for comparison of the contents of register pairs p1 and p2. The addresses (p1 and p2) can have the values 0, 2, 4, 6, 8, A, C or E; odd value is interpreted as the next lower value. AR is not affected. Zero is one-set if the contents of the two register pairs are equal (otherwise zero-set) and carry is one-set if the contents of register pair p2 are less than or equal to p1 (otherwise zero-set). Note that one byte in the stack is used during execution of 43 and 48.

45, SHRI
44, SHLE

The contents of AR can be shifted one step to the right (by 45, SHRI) and one step to the left (by 44, SHLE). Zero is one-set if the value in AR becomes zero (otherwise zero-set). Carry is set according to the bit that is shifted out. In 44 the least significant bit is set to zero and in 45 the most significant bit is set to zero.

4248 ROLE
4258 RORI

It is also possible to rotate the contents of AR one step left by 4248 ROLE and right by 4258 RORI. These instructions are performed in the same way as 44 and 45 but carry is shifted in. Thus, in 4248 carry is entered into LSB, and in 4258 carry is entered into MSB. Zero and carry are set as in 44 and 45.

Memory reference instructions

5, LIN
LDE
SIN
SDE

The instructions with OP 5, format 2 are used in connection with storing in or loading from the RWM. The argument contains a) the address of the register pair that holds the 16-bit address to the memory b) indication whether the contents of AR shall be stored in the memory, or AR shall be loaded from the memory and c) indication whether the memory address in the register shall be incremented or decremented by 1 after the memory access. See table 3 in Appendix 1.

During execution of these instructions one byte of the stack is used. Zero is one-set if the contents of the register pair become zero. (otherwise zero-set) and carry is one-set if carry is obtained in connection with incrementing or decrementing (otherwise zero-set). (In connection with decrementing, carry is obtained always except when the contents of the register pair become FFFF.)

From the table (3 in appendix 1) it is evident that a "1" in the most significant position of the argument indicates that the contents of AR shall be stored in the memory. A "1" in the least significant position indicates that the memory address shall be decremented after the access. The two bits in the centre positions select the register pair.

I/O-instructions

6 and 7

The instructions with OP 6, format 1 are output (write) instructions and the instructions with OP 7, format 1 are input (read) instructions. See table 1 in Appendix 1, which shows the most common I/O-instructions. Other I/O-instructions exsist in certain applications.

Write instructions transfer data from AR (8-bits), together with the parity bit (odd number of ones) that is generated in the MPU, to output units. AR, C and Z are not affected.

Read instructions transfer data from input unit to AR (8 bits). Any parity bit is transferred to carry flip-flop, if it is not used carry is zero-set. Zero flip-flop is one-set if the content of AR becomes zero (otherwise zero-set).

INTERRUPT SIGNAL SYSTEM

The MPU operates in general with a low priority program. When an event occurs that calls for action by the MPU, the program is interrupted and the program that is called for starts. The interrupt signals are arranged in priority order. The interrupt logic works on three levels 0, 1 and 2. An interrupt signal to level 0 causes the program working on levels 1 or 2 to be interrupted. An interrupt signal to level 1 does not interrupt a program on level 0 or 1 but interrupts a program on level 2. Within levels 0 and 1 there are four sublevels. Interrupt signal to a sublevel of the same level as the current working program does not cause interrupt. (The interrupt signals listed below are only examples.)

Level	Sublevel	Interrupt address	Interrupt signal
0	0	0000	Power on/reset
	1	0002	MCA I rec. ready
	2	0004	MCA I transm. ready
	3	0006	EBA interrupt

Level	Sublevel	Interrupt address	Interrupt signal
1	0	0008	Clock interrupt
	1	000A	Keyboard interrupt
	2	000C	GEA interrupt
	3	000E	PRA interrupt

The interrupt priority logic contains one flip-flop for each interrupt signal. These flip-flops can be set independently of each other. When an interrupt flip-flop is set, an interrupt program will be started if

- No program is working at a higher level
- No program is working at the same level
- Interrupt is permitted after current instruction

Interrupts are prohibited if a) instruction 4B, EXTN has been given and has not yet been followed by a jump or call instruction and b) instruction 4D, DITR has been given. The interrupt flip-flop remains set until the interrupt is made.

If another interrupt signal to the same sublevel is received without the first interrupt signal having caused an interrupt, only one interrupt will be made.

When an interrupt program is to be started, the next instruction of the current program is inhibited. Instead the stack is loaded with three bytes containing current value of instruction counter (i.e. the address to next instruction in the program that will be interrupted), the value of the two condition flip-flops (C and Z), the value of the register block flip-flop and finally the contents of accumulator register (AR). Then the interrupt address is entered into the instruction counter. The register block flip-flop is zero-set before the interrupt program starts.

The instruction received from the interrupt address (the most significant bits are zero) is a jump instruction. This instruction causes a jump to be made to the address where the interrupt program starts.

During the interrupt program, its priority is stored to allow the MPU to know if another interrupt signal is to be executed or not. The priority is set when the interrupt address has been entered into the instruction counter. At the same time, the interrupt flip-flop is reset to allow another interrupt on same sublevel immediately.

When the interrupt program is finished, the MPU checks whether other interrupt signals (on the same or lower level) have arrived. If so, these interrupt signals are handled in priority sequence.

Description	Mnemonic	First byte: OP (4 MSB)	Argument (4 LSB) 0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Second byte
Absolute jump	JUMP	0	Used as MEMA:8 – MEMA:11																MEMA:0–MEMA:7
Absolute jump to subroutine	CALL	1	Used as MEMA:8 – MEMA:11																MEMA:0–MEMA:7
Relative jump on true condition	JCTc	2	Zero (Z)	Carry (C)	AR data:7	AR + C odd	MCA error	MCA break	MCA overrun	EBA status:1	EBA status:2	DIA repeat	DIA ID char	DIA overrun	Spare	Printer operable	Spare	Rel. jump to subroutine	Always true	Relative address
Relative jump on false condition	JCFc	3	Zero (Z)	Carry (C)	AR data:7	AR + C odd	MCA error	MCA break	MCA overrun	EBA status:1	EBA status:2	DIA repeat	DIA ID char	DIA overrun	Spare	Printer operable	Spare	Rel. jump to subroutine	Always true	Relative address
		4	BLK0 Select register block 0	BLK1 Select register block 1	Multiple byte instr., see sep. table	MORRab Move reg b to reg a	SHLE Shift AR left	SHRI Shift AR right	PUSH Put AR to stack	POPP Load AR from stack	DRCM Compare p1 and p2	GRES General reset	ARIC IC+AR to IC	EXTN Change IC:12	ENTR Enable interrupt	DITR Disable interrupt	CPRM Change program	RTRN Ret fr subrout		Exists at 42, 43 and 48 43: Address to a (4 MSB) and to b (4 LSB) 48: Address to p1 (4 MSB) and to p2 (4 LSB)
Memory reference instructions, see separate table		5	LIN0	LDE0	LIN2	LDE2	LIN4	LDE4	LIN6	LDE6	SIN0	SDE0	SIN2	SDE2	SIN4	SDE4	SIN6	SDE6		—
Output instructions		6	UTPA Address to PA	UTPR Data to printer	UTDI Status to display	UTDA Address to DA	UTED Data to EBA	UTES Status to EBA	Spare	Spare	UTMM Write transmitter	UTSR Set req. to send	UTRR Reset req. to send	UTSY Look for SYN	UTGC Order to GEA	UTGD Data to GEA	Spare	Spare		—
Input instructions		7	INPA Address from PA	INDA Address from DA	INKB Data from KB(ID)	INSO Status from sg1	INBS Data from EBA	Spare	Spare	Spare	INMM Read receiver	Spare	Spare	Spare	INGD Data from GEA	INGC Status from GEA	Spare	Spare		—
Load AR from register a	LDRa	8	Used as address to register																Use next byte instead of register	Data to be used in logic and arithmetic operations instead of register contents at instructions 8–E
AR · a to AR	ANDa	9	Used as address to register																	
AR + a to AR	IORa	A	Used as address to register																	
AR ⊕ a to AR	EXOa	B	Used as address to register																	
Add AR and a; result to AR	ADDa	C	Used as address to register																	
Sub a from AR; result to AR	SUBa	D	Used as address to register																	
Compare AR and a	CMPa	E	Used as address to register																	
Store AR in register a	STRa	F	Used as address to register																No op	—

TABLE 1. Instruction List

Second byte: 4 MSB	4 LSB 0	1	2	4	5	6	9	B	Third byte	Fourth byte
Used as address to register (a) or register pair (p, p+1)	Increment reg a by 1	Decrement reg a by 1	Load reg a with 3rd byte	Inc reg pair p by 1	Dec reg pair p by 1	Load reg p with 3rd, reg p+1 with 4th byte	Add AR and a and C; result to AR	Sub a from AR plus C; result to AR	Data used in 3 and 4 byte instructions	Data used in 4 byte instructions
	SRINa[1]	SRDEa[1]	SRLDa[1]	DRINp	DRDEp	DRLDp	ADCYa[2]	SBCYa[2]		

[1] a = 0–F

[2] a = 0–E

[2] a = F: use data instead of a

4248 ROLE Rotate AR left
4258 RORI Rotate AR right

TABLE 2. Instructions with first byte 42

		Pair of registers (MSB in even reg) 0,1	2,3	4,5	6,7
Load and increment	(LIN)	50	52	54	56
Load and decrement	(LDE)	51	53	55	57
Store and increment	(SIN)	58	5A	5C	5E
Store and decrement	(SDE)	59	5B	5D	5F

TABLE 3. Instructions with 4 MSB = 5 (0101)

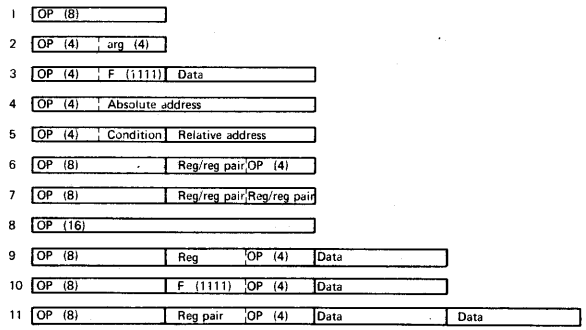


TABLE 4. Instruction formats

SWI

STANSAAB

Alfaskop 3500 Data Terminal System

Technical Description

Control Unit 3501

Introduction

Mechanical Design

Brief Outline

Cluster Operation

Timer

Multiplexor

Display Adapter 3532

Printer Adapter 3533

Micro Processing Unit

Memory

Modem Communication Adapter

External Bus Adapter

Line multiplexor

EE557–412
1975–01–10

Introduction

This part describes all standard units that can be included in the Control Unit 3501. The basic equipment of CU 3501 incorporates one PC-board mounting frame and one power supply. Logic units controlling up to eight display adapters and up to eight data printers can be inserted in the first frame. If the capacity is to be expanded, up to three additional frames (Control Module 3503 or Expansion Module 3531) can be mounted.

The following PC-boards are included in all frames (3501, 3503, 3531)

- MPU, Micro processing unit (two PC-boards)
- MEM, Memory (one or two PC-boards)
- MUX
- TIM
- DA, Display adapter (one PC-board for each display unit)
- PA, Printer adapter (one PC-board for each printer)

In the basic frame of CU 3501 also a modem communication adapter (MCA) PC-board is included. Moreover, if an expansion module is added, an external bus adapter (EBA) PC-board must be included in the basic frame.

A modem communication adapter also is incorporated in the Control Module 3503. If a CM 3503 is added, a Line Multiplexer 3534 must be included in the Control Unit 3501.

The Expansion Module 3531 also includes an external bus adapter PC-board.

The basic frame of control unit and the control modules work quite independently of each other.

In control units equipped with expansion modules, the frames are connected via an external data bus. The modem communication is completely controlled by the MPU in the basic frame. This MPU also acts as a master as regards the data exchange across the external data bus. Editing in connection with communication with the computer is made in the master frame.

Mechanical Design

The basic equipment of Control Unit 3501 includes a cabinet with doors at both sides to facilitate access for servicing etc. The rack in the cabinet can accomodate up to four PC-board mounting frames and up to four power supplies. The PC-board mounting frame is provided with a PC interconnection board onto which all connectors are soldered. 20 PC-board positions are available in a mounting frame.

There are two sizes of PC-boards, one occupying the full height of the mounting frame (full board), the other occupying only half the height of the frame (half board). The PC-boards have printed circuits on both sides. The full board has two 82 polar edge connectors. For the voltage supply there are bus bars on most of the PC-boards to decrease the resistance in the conductors.

The logic circuits are built up primarily using MSI circuits but also SSI and a few LSI circuits are used as well as discrete components.

A fan blower unit is used to cool each PC-board frame and each power supply. On top of one of the memory PC-boards a sensor for thermal release is mounted. If the temperature becomes too high the +5 V supply to this frame is disconnected.

The power supply gives the voltages +5 V, +12 V, −9 V and −12 V. The latter three are built up with rectifiers, stabilizing circuits and series regulating transistors. The +5 V supply is stabilized in a switching regulator, whereby high effeciency is yieldded. The regulator can be of two different makes: Advance and Triolab. If the latter is used a low-pass filter is connected in series at the input.

The mains power is distributed via a contactor in the connection terminal assembly at the bottom of the cabinet.

Timer

Contents

DETAILED DESCRIPTION 1

- ROM 1, 2 and counters, sh 3 1
- ROM programs 3
- Condition selector 6
- Decoder logic, sh 4 6
- Last bit generator, sh 4 6
- DIM address control, sh 4 7

FIGURES

1. Program time diagram 3
2. Position program flow diagram 4
3. Main program flow diagram (24 lines x 80 characters) 5

DETAILED DESCRIPTION

Note: *The following paragraphs deal with a program controlling a screen of 24 text lines of 80 characters each, a command of nine bits, five tag bits and nine KB/ID/LP bits. Other screen set-ups may occur.*

The timer, TIM, contains the logic to control eight display adapters, DA, within one frame the memory address counter in MUX and the blocking signals to MPU to prevent this from using the display memory, DIM, when it is occupied by the line store in DA.

Each sweep the following signals are generated:

Horizontal hold

pos 10, 15, 21, 53, 85, 100 and 101

Each frame the program in TIM generates a train of events to define the different parts of the frame which is shown on page 3.

TIM contains the following main blocks:

• ROM 1 and ROM 2	IC7 and 6, sh 2
• Position program counter	IC20, sh 2
• Main program counter	IC9 and 30/3, sh 2
• Main address register	IC68, sh 2
• Address selector	IC8, sh 2
• Column counter with oscillator	IC56, sh 2
• Position, sweep and line counters	IC19, 43 and 31, sh 2
• ROM 2 decoder logic with latch circuits	IC49, 50, 37, 38 and 25, sh 1
• Condition selector	IC3, sh 1
• Last bit generator	IC53, sh 1
• DIM address control	sh 2

ROM 1, 2 and counters, sh 3

The presentation timing is mainly controlled by ROM 1 and 2, IC7 and 6, which control a set of counters as well as provide proper output control signals. The addresses for the two ROM are generated in two counters, the position program counter, IC20, and the main program counter, IC9 and 30/3.

Two different programs are stored in ROM and the activating circuits IC45 decides which to use by switching the address selector, IC8, into proper state, i.e. connecting one of the counters to the

ROM inputs.
The value of the main program counter is stored in the main address register, IC68, to ensure safe function when the program provides a jump.

Both ROM get the same address, but their outputs are used for different purpose. ROM 1 controls the main program counter, the position counter, the sweep counter and the line counter. The output of ROM 1 is divided into two parts. Bits 0, 1 and 2 are applied to the load decoder, IC32, which converts the value into a proper load pulse:

	Bits 2	1	0	
Load position counter	0	0	0	LPC
Load sweep counter	0	0	1	LSC
Load line counter	0	1	0	LLC
Load main address unconditional	0	1	1	LMA
Load main address if condition selector replies NO	1	1	1	

These signals thus indicate which counter should be loaded. Bits 3 to 7 is the value the counter at hand should be set into.

The counters as well as other circuits in TIM are reset by General Reset via 66/11, 52/11 and 17/11 and 6, sh 3.

The column counter in TIM serves as a master counter for each DA column counter. The DA column counters are synchronized every column 9 from TIM. TIM column counter also steps the position counter by column 9 and after a certain number of step-pulses, the position counter provides carry, CPC.
The number of step-pulses before carry depends on the value the position counter was set into by ROM 1. When CPC appears, the activating circuits feeds low level from 39/11. Thereby the position program counter feeds a new address to the two ROM via the address selector, IC8. ROM 1 thus feeds a new value to the four counter inputs and a load value is fed to the load decoder, IC32.
The load decoder is strobed by the signal GTP to ensure that the ROM output is stable before decoding is made. The decoder then feeds LPC and the position counter is set in parallel with bits 3 to 7 from ROM 1. The sweep counter, the line counter and the main address counter are controlled in a similar way. The activating circuits are in these cases set to main address, i.e. 39/11 feeds high level.
The main program counter is only loaded (LMA) when a jump is made. This is determined by ROM 1 or the condition selector.
Thus each time a carry appears from a counter, ROM 1 and 2 gets a new address. ROM 2 thereby feeds 6 bits to the decoding logic, which detects the meaning of each carry.
Consider the following two examples:

- Assume that the sweep counter has fed carry (CSC) at the end of the command byte (sweep 9). This carry signal causes a step pulse for the main program counter, which feeds a new address to ROM 1 and 2. ROM 1 thereby indicates "set sweep counter to count five sweeps" (which means the five sweeps of the tag byte). ROM 2 indicates "reset byte 0 (command byte) and set byte 1 (tag byte)". The result of this is that the sweep counter is stepped five times and then produces carry to get a new setting and the decoder logic feeds byte 1 to DA during these five sweeps.

- Assume that the position counter has fed carry (CPC) at position 10. The carry signal steps the position program counter, which feeds a new address to ROM 1 and 2. ROM 1 indicates "set position counter to count five positions" and ROM 2 indicates "set position 10". The position counter is stepped five times from the column counter and then produces a new carry. The position program counter is stepped by the carry signal and a new address is applied to ROM 1 and 2.
  ROM 1 indicates "set position counter to count six positions (to pos 21)" and ROM 2 indicates "set pos 15".

  The program also decides when to set and step the line counter. It is stepped by the signal CUL, mainly generated by carry from the sweep counter at the end of a text line cycle.

ROM programs

The logic in TIM must each frame provide a train of events. To control this, two programs are stored in ROM 1 and 2. Note that the ROM work in parallel and thus the two programs are not distinguished by ROM 1 and 2.

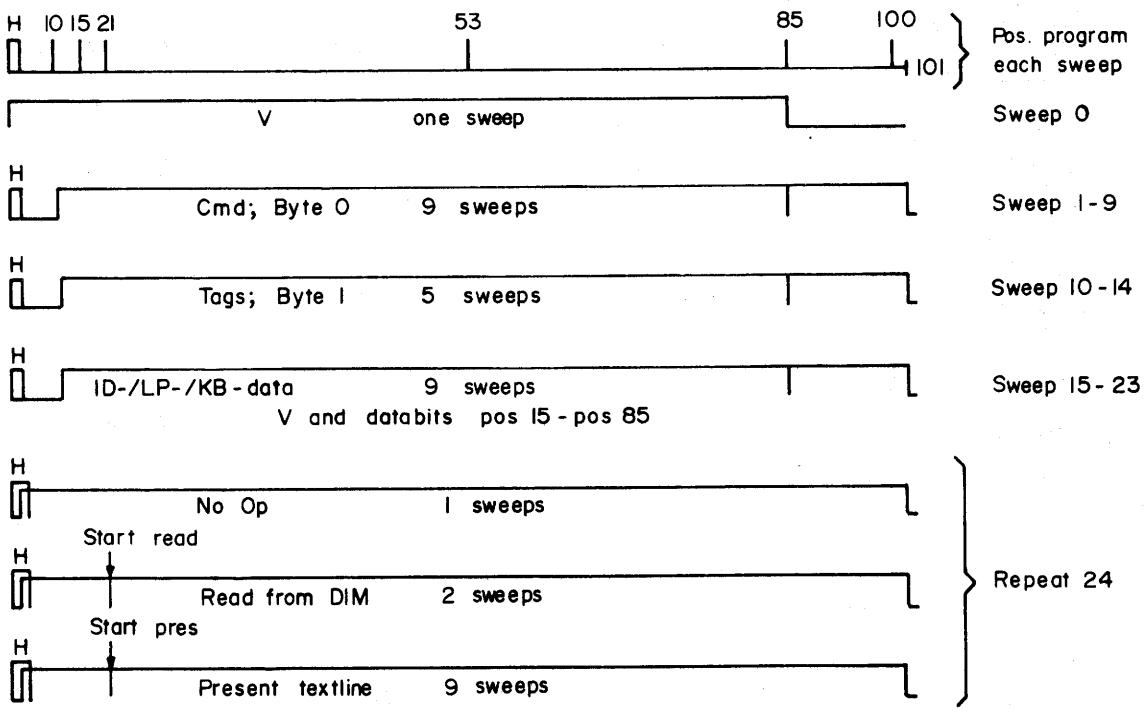
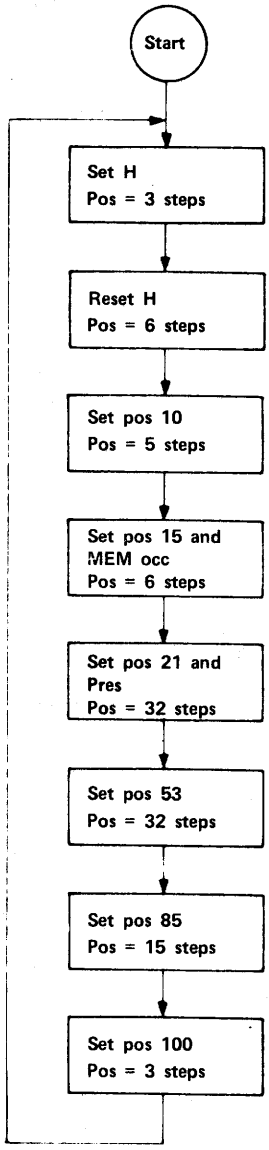


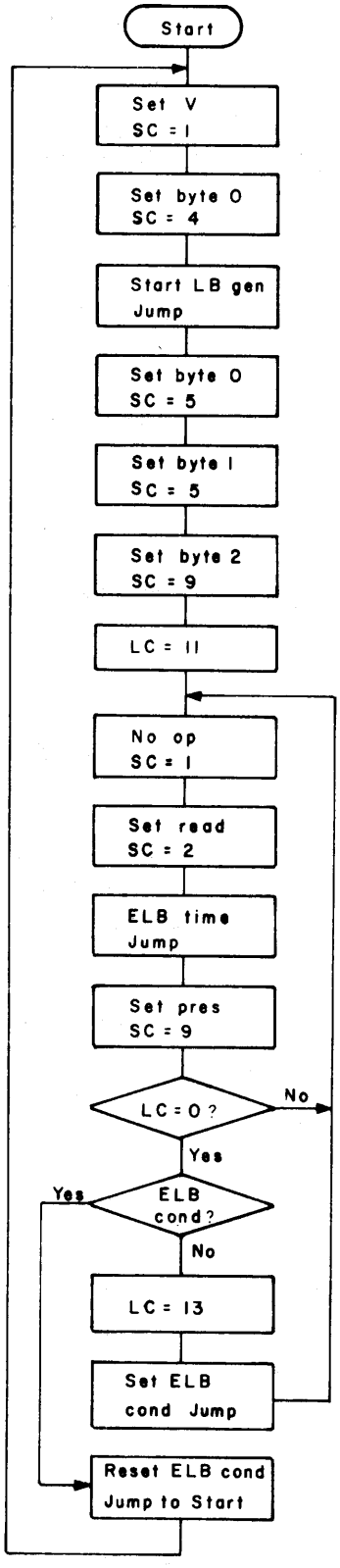
Figure 1. Program time diagram

Figure 2 shows the position program, which is stored in the last eight addresses in ROM, i.e. address bits 4 and 3 are both ones. Note the ROM 1 output bits 0–2 is 000 which means "load position counter". The position program is activated when CPC appears between column 1 and column 5 from the column counter during the particular position.



Address	ROM 1		ROM 2
4 3 2 1 0	7 6 5 4 3 Counter value	2 1 0 LPC	5 4 3 2 1 0 Event
1 1 0 0 0	1 1 1 0 0	0 0 0	0 1 1 0 0 0
1 1 0 0 1	1 1 0 0 1	0 0 0	0 1 1 1 0 0
1 1 0 1 0	1 1 0 1 0	0 0 0	0 0 0 0 0 0
1 1 0 1 1	1 1 0 0 1	0 0 0	0 0 0 1 0 0
1 1 1 0 0	1 1 1 1 1	0 0 0	0 0 1 0 0 0
1 1 1 0 1	1 1 1 1 1	0 0 0	0 0 1 1 0 0
1 1 1 1 0	1 0 0 0 0	0 0 0	0 1 0 0 0 0
1 1 1 1 1	1 1 1 0 0	0 0 0	0 1 0 1 0 0

Figure 2. Position program flow diagram



Address	ROM 1		ROM 2
4 3 2 1 0	7 6 5 4 3	2 1 0	5 4 3 2 1 0
	Counter value	Load	Event
0 0 0 0 0	0 1 1 1 0	0 0 1	1 0 0 0 0 1
0 0 0 0 1	0 1 0 1 1	0 0 1	1 0 0 0 1 0
0 0 0 1 0	0 0 0 1 0	0 1 1	0 1 0 1 0 1
0 0 0 1 1	0 1 0 1 0	0 0 1	1 0 0 0 1 0
0 0 1 0 0	0 1 0 1 0	0 0 1	1 0 1 1 0 1
0 0 1 0 1	0 0 1 1 0	0 0 1	1 1 1 1 0 1
0 0 1 1 0	0 0 1 0 0	0 1 0	1 0 0 0 0 0
0 0 1 1 1	0 1 1 1 0	0 0 1	1 0 0 1 0 1
0 1 0 0 0	0 1 1 0 1	0 0 1	1 0 1 0 1 0
0 1 0 0 1	0 1 0 0 1	0 1 1	0 1 1 0 0 1
0 1 0 1 0	0 0 1 1 0	0 0 1	1 0 1 1 1 0
0 1 0 1 1	0 0 1 1 0	1 1 1	0 0 0 1 1 1
0 1 1 0 0	0 1 1 1 0	1 1 1	0 0 0 0 1 1
0 1 1 0 1	0 0 0 1 0	0 1 0	1 0 0 0 0 0
0 1 1 1 0	0 0 1 1 0	0 1 1	0 1 0 0 1 0
1 0 0 0 0	0 0 0 0 0	0 1 1	0 1 0 1 1 0

Figure 3. Main program flow diagram (24 lines x 80 characters)

Condition selector

The condition selector, IC3 sh 1, decides whether or not a conditional jump is to be made in the program. The selector checks two conditions: Carry line counter and Enable last bit, ELB condition. ELB will be explained further on in the paragraph Last bit generator. The condition selector is enabled from the group matrix, IC1, when bits 0 and 1 from ROM 2 are ONE's. Bits 2 and 3 decides which condition to be tested. The output of the selector provides YES: "step main address" or NO: "jump according to ROM 1". See program flow diagrams.

Decoder logic, sh 4

The five bits b_4 to b_0 from ROM 2 are applied to the decoder logic, which thereby indicates proper "event".

The logic contains:

- A group matrix, IC1, which enables the proper latch circuit as well as the condition selector.
- A subject matrix, IC13, which selects a certain latch within the enabled group.
- A position latch circuit, IC49, directly subject addressed from ROM 2.
- Four main program latches, IC50, 37, 38 and 25, which also contain the H-hold condition of the position program.

Setting of the latches is made according to "instruction" lists of ROM 2, shown together with the program flow diagrams. In order to define the last position of the sweep, pos 101, position 100 provides set condition for 46/3 and column 1 contains the clock pulse. The FF is cleared by column 7.

Note that IC50, 37, 38 and 25 are set by low level on the S-inputs and reset by low level on the D-inputs. R-input clears all latches.

The decoder logic also generates the signals count up sweep, CUS, and count up line, CUL.

The following times are valid

- Presentation starts in pos 22/column 3 and ends in 101/9
- Display condition, D, starts in 21/4 and ends in 101/3
- H-sync starts in 1/4 and ends in 4/3

Last bit generator, sh 4

The last bit generator, IC53, feeds set pulses for DA status register at the end of the tag byte and for DA data register every nine sweeps during the succeeding 144 sweeps. Normally the data register is set only once (during sweep 23) every frame. However, in certain systems all 16 ID-bytes are transferred during one frame (16 bytes · 9 sweeps = 144) and therefore TIM program provides these 16 set pulses each frame. See main program flow diagram.

The last bit generator is started by the program from latch 25/13 by the signal enable last bit, ELB. This signal sets 0111 (7) into the counter, which is clocked by pos 100. When nine clock pulses have arrived, the counter provides carry. The carry signal is gated with byte 1 (tags) and used as a set pulse for DA status register. Carry is also used as a load pulse for the counter, which thereby is set to 0111 again.
After nine more clock pulses a new carry appears, which this time is used as set pulse for DA data register. The cycle is repeated 15 times more and then ELB is switched off by the conditions reset ELB condition, 38/4, and ELB-time from the program. The signals are gated in 73/3, which resets the ELB latch 25/13.

DIM address control, sh 4

Each time a line shall be generated on the screen, TIM provides the loop: No Op, Read, Present. At the beginning of the two read sweeps, the signal DIM to Line Store is fed from 39/6, sh 4. This signal sets the (8) line stores in the DA's into read mode and switches the address selector in MUX so that 11 bits from memory address counter, MAC, are applied to DIM address inputs. DIM to LS also releases the Read CP FF 35/12, which is clocked from the Pres CP FF 36/11. Thereby the presentation cp is divided by two into the read cp. The Read CP FF thus provides Cp for both line store (P2/8) and for MAC (P1/85) and this means that characters are entered from DIM and into LS with the rate of 40 characters/sweep.

DIM to LS appears between pos 21 and 101 during both read sweeps and thus a gap is formed in the signal between pos 101 of the first sweep and pos 21 of the second. During these two sweeps it is necessary to prevent MPU from addressing any of the DIM (and PRM) and TIM program therefore provides the condition MPU stop condition from 39/3, which is fed to 59/9 as opening condition. Should MPU try to address any DIM or PRM during "DIM to LS", this is detected by MUX and the signal DA or PA MEMA decoded is fed via P1/86 to 59/10. 48/3 is thereby cleared and MPU write signal is blocked in 47/10. Furthermore 78/12 is cleared thereby opening the tri-state 58/3 to feed the signal Stop Cp Counter to MPU. As soon as the stop condition to 59/8 disappears, MPU is released and can address DIM and PRM again.

When the two read sweeps are finished DIM to LS disappears and thus 35/12 is blocked. When on the following nine sweeps the display condition, D, is valid, Cp line store is fed from 36/9 via 34/ /11.

TIM can be equipped with a program controlling a frame of 12 lines 40 characters. In this case only 40 step pulses must be fed to MAC during the read phase. However, since the line stores have 80 positions, 80 Cp line store must be fed to DA.
Every other line store clock pulse is therefore fed from 35/5. This means that the line store is loaded e.g. AA, BB, CC etc.

Multiplexor

Contents

DETAILED DESCRIPTION 1

- DIM and PRM addressing logic 1
- MUX DA channel 1
- MUX PA channel 3

DETAILED DESCRIPTION

DIM and PRM addressing logic

During read phase (two sweeps every text line) DIM and PRM are addressed from the 11 bits memory address counter, MAC, IC62, 50 and 38. MAC is set into 7FF by vertical hold and pos 9 in the beginning of each frame, and since memory bus is low level significant, this is considered as 000 by DIM and PRM. Each time the two read sweeps are performed, the signal DIM to line store from TIM sets the address selector, IC61, 46 and 37, into the state where MAC is connected to the DIM address bus, DIMA and DA line store is set into read mode. TIM then clocks MAC 80 times (count down) and thereby 80 characters form DIM is read into DA line store, which is clocked simultaneously. During the read phase, MPU is blocked from reading or writing into any of DIM or PRM, see section TIM and section Cluster operation.

When the signal DIM to line store is not valid, the address selector connects the 11 LSB of MPU memory address bus, MEMA, to DIM address bus, DIMA. The five MSB of MEMA are always connected to two enable decoders, IC73, controlling DIM and IC75, controlling PRM. These enable decoders selects one of the 16 possible DIM and PRM depending on the five MSB of MEMA. These bits are organized:

	15	14	13	12	11
DA1	0	1	0	0	0
DA8	0	1	1	1	1
PA1	1	0	0	0	0
PA8	1	0	1	1	1

Bits 14 and 15 thus decides whether a DA or a PA is enabled and bits 11–13 which one.

Note: Since MEMA is low level significant the decoded figure 7 is considered as 0 and 0 as 7 in decoders IC73 and 75. I.e. the output enable signals are defined reversely to the manufacturers.

The two decoder enable signals from 26/6 and 14/3 are gated in 26/8 to provide the signal PA/DA address decoded, which is fed to TIM. This signal is used in TIM to stop cp counter in MPU and to block MPU write signal during the read phase.

MUX DA channel

Example of DA/MPU data exchange

Consider the example in section Cluster operation where it is assumed that the operator at display 3 enters the letter "K" at line 15 pos 42.

The DA channel in MUX is internally controlled by the clock pulse generator, CP generator, IC27 and 15, which is clocked by CP/2 from TIM, i.e. the rate of the column counter oscillator. The

CP generator provides the eight pulses CP0 to CP7.

DA input address counter is stepped by step DA input address, SIA, containing CP0 gated by the stop FF5/8. The counter is connected to the enable selector, IC64, which in turn is connected to the enable decoder, IC81. As the address counter is stepped each DA is enabled, one at a time from the enable decoder. When, at a certain time, the address counter reaches "display 3", the or-wired DA interrupt signal is detected (P1/33). At the succeeding clock pulse CP2 the DA interrupt FF4/12 is set, provided that 28/12 is not blocked.

Clock pulse CP3 then sets the read status FF17/3 and CP5 provides set status which enters the content of DA number 3 status register into MUX DA status register, IC55. DA status register is read by the signal DA status to MUX from 17/5. Clock pulse CP6 then clears 17/3 and clocks KB irpt FF, 17/11, which feeds KB irpt to MPU. 17/11 is cleared by the succeeding CP1. Clock pulse CP7 clocks the stop FF5/11 and since 4/12 is set 5/11 will be set to ZERO. The stop signal from 5/8 thereby blocks CP0 and CP3. This means that DA input address counter is stopped (indicating display 3).

At this state MUX status register contains the status of DA3, DA input address counter is stopped at display 3 and MUX is idle awaiting response on its KB interrupt.

When MPU is ready to fetch status from MUX, the instruction INDA is fed to MUX via I/O address bus. The instruction decoder, IC34, thereby feeds low level to 31/8, which blocks the tri-state buffer IC43, 44 and 32 and enables the selector, IC67 and 68. The decoder also sets the selector to feed MUX status register and DA input address counter content to the I/O data bus. MPU then performs disable interrupt and then feeds INKB to MUX instruction decoder. The decoder signal INKB blocks the tri-state buffer except 32/3 which feeds the least significant bit of the DA bus to the I/O data bus. The decoder signal also connects DA bus to the I/O data bus via the selector (and 32/3) and feeds Data from KB (INKB) to DA in order to open the tri-state outputs of the DA data register. At the trailing edge of INKB (34/11 rises to high level) 5/3 is set, which in turn sets DA irpt FF, 4/12, to ZERO at the next CP2. The succeeding clock pulse CP7 then resets the stop FF5/11, and thus CP0 is no longer blocked in 16/4. This means that DA input address counter receives step pulses each CP0 and thereby starts to scan the DA for a new DA interrupt.

At this state MPU has fetched DA status, DA address and KB data. When this is handled by MPU and set into DIM, MPU must set a "click" command into DA command register.
MPU thus feeds UTDA on the I/O address bus to MUX instruction decoder and at the same time display 3 address on the I/O data bus. The instruction decoder thereby feeds set pulse to DA output address register, IC52, which is set with the address on the I/O data bus ($b_1 - b_3$). At the same time 7/3 is preset and this sets the stop FF5/11, at the succeeding CP7, thus blocking step DA input address. Furthermore 53/3 is set which blocks DA interrupt in 28/12 and sets the enable selector towards the DA output register.

MPU then feeds the instruction UTDI on the I/O address bus and at the same time the command byte containing the "click" bit as ONE on the data bus.
The command byte is fed to the DA bus via the tri-state buffer and the decoded UTDI is fed as a set pulse for the command register in DA via MUX 32/11. The trailing edge of the low level signal from 34/5 sets 7/3 to ZERO thereby releasing the stop FF which will be set the succeeding clock pulse CP7. CP0 can then step DA input address counter again and at the succeeding CP3, 53/3, is set to ZERO and 28/12 is then opened for another DA interrupt.

MUX PA channel

Also see MUX DA channel.

The PA channel is similar to the DA channel. The internal timing is taken care of by the clock pulse generator mentioned earlier. The clock pulse CP0 provides stepping of the PA input address counter, IC60, which is connected to the enable logic, IC70 and 82. The address counter thus scans the connected PA for PA interrupt by enabling the various PA one at a time.
When a PA interrupt is detected (P1/67), the PA interrupt FF4/9 is set at the succeeding CP2. The high level from 4/5 then sets 8/3 at CP3 and resets the stop FF20/3 at CP7. CP6 clears 8/3 and between CP3 and CP6, printer interrupt is fed to MPU as well as a read pulse (PA status to MUX) for the status from PA. The status is set into MUX status register, IC59, by CP5.
Since the stop FF20/3 is reset the step pulse (CP0) for the PA input address counter is blocked in 16/10 and further setting of status in 22/10.

Thus Printer interrupt is fed to MPU and status and PA address is available in PA status register and PA input address counter. MPU then fetches address and status by the instruction INPA to the instruction decoder, IC34. The low level signal from 34/9 blocks the tri-state buffer, IC47, 48 and 36, and feeds status and address to the I/O data bus via the selector, IC71 and 72. At the trailing edge of INPA (34/9 rises to high level) 8/11 is set (provided that strapping plug 12 is strapped 13–4). 8/11 then releases the control logic and PA input address counter starts to scan the PA for further PA interrupt. Note that MUX is prepared for a printer which can transmit data (ASR). IC35 can decode an instruction meaning input data from printer (INPR).

In the meantime MPU investigates status and address from MUX and decides that a character should be transferred to a certain PA transmitter buffer. MPU then fetches next character by addressing correct PRM via DIM/PRM addressing logic.
Thereafter MPU feeds the instruction UTPA on the I/O address bus and at the same time PA address on the I/O data bus. UTPA is decoded in IC34 and low level is fed from 34/7. Thereby 7/11 is preset, which in turn resets the stop FF20/3 at the succeeding CP7. Thus CP0 is blocked and PA input address counter stops. The outputs from the stop FF and 7/9 provides set condition for 20/11, which is set by the succeeding CP3. Thereby PA interrupt is blocked in 22/1 and PA enable selector connects PA output address register to the enable decoder. The decoded signal UTPA has previously set PA address into the output register from I/O data bus. Correct PA is now enabled and MUX PA channel is idle since the control circuits are blocked and PA input address counter is stopped.
MPU then feeds UTPR on the I/O address bus and at the same time the character to be printed on the I/O data bus. UTPR is decoded in IC34 and a low level from 34/6 is fed via 1/8 to PA transmitter as a set pulse. Since the tri-state buffer between I/O data bus and PA bus is open, the character at hand is set into the transmitter buffer of the enabled PA.
The trailing edge of UTPR (34/6 rises to high level) will reset 7/11 and thereby the control logic is released and PA input address counter starts to scan the PA for further interrupt.

MUX PA channel also holds a frequency divider IC24 and 2 which is fed with 1 200 and 300 Hz from MPU. MUX then feeds 1 200, 300, 109 and 75 Hz as CP for PA.

Display Adapter

Contents

DETAILED DESCRIPTION 1

- Display Memory, DIM 1
- Line store 3
- Character generator 3
- Video register 5
- Column counter 5
- Field control logic 5
- Data transfer circuits 10
  - Command byte 10
  - Tags and status 10
  - Interrupt and overrun 11
  - Data byte 11

FIGURES

1. Display memory 1
2. CG code table 4
3. Character layout 6
4. Underline of the word "opt" 7
5. Underline time diagram 8
6. Intensity control time diagram 9
7. Intensity control with line wrap-around 9
8. DA block diagram 12

DETAILED DESCRIPTION

Display memory, DIM

See also MUX and section Cluster operation.

Each display memory, DIM, contains: 16 RWM dips, IC49–56 and IC63–70. Each dip holds 1024 bits (one kbits) and the dips are arranged in two groups of eight dips each. Each group thus holds one kbytes, which is addressed with ten address bits. The two groups are selected by an eleventh address bit, which is connected to the chip enable (CE) inputs. Thus when bit 10 has low level IC49–56 are enabled and IC63–70 blocked from 76/10 and when bit 10 has high level, IC63–70 are enabled. If the address from MUX is defined as high level: 7FF is byte ZERO and $000_{(16)}$ is byte 2048. This means that the character in position 0, line 0 is stored in address $7FF_{(16)}$ and the character in position 80 line 24 (24 x 80 = 1920) in $07F_{(16)}$. However, the address bus is usually considered low level significant, which is obvious in the table concerning DIM organization.

Note: There are no inverters between the connector and the RWM inputs as in the memory PC boards.

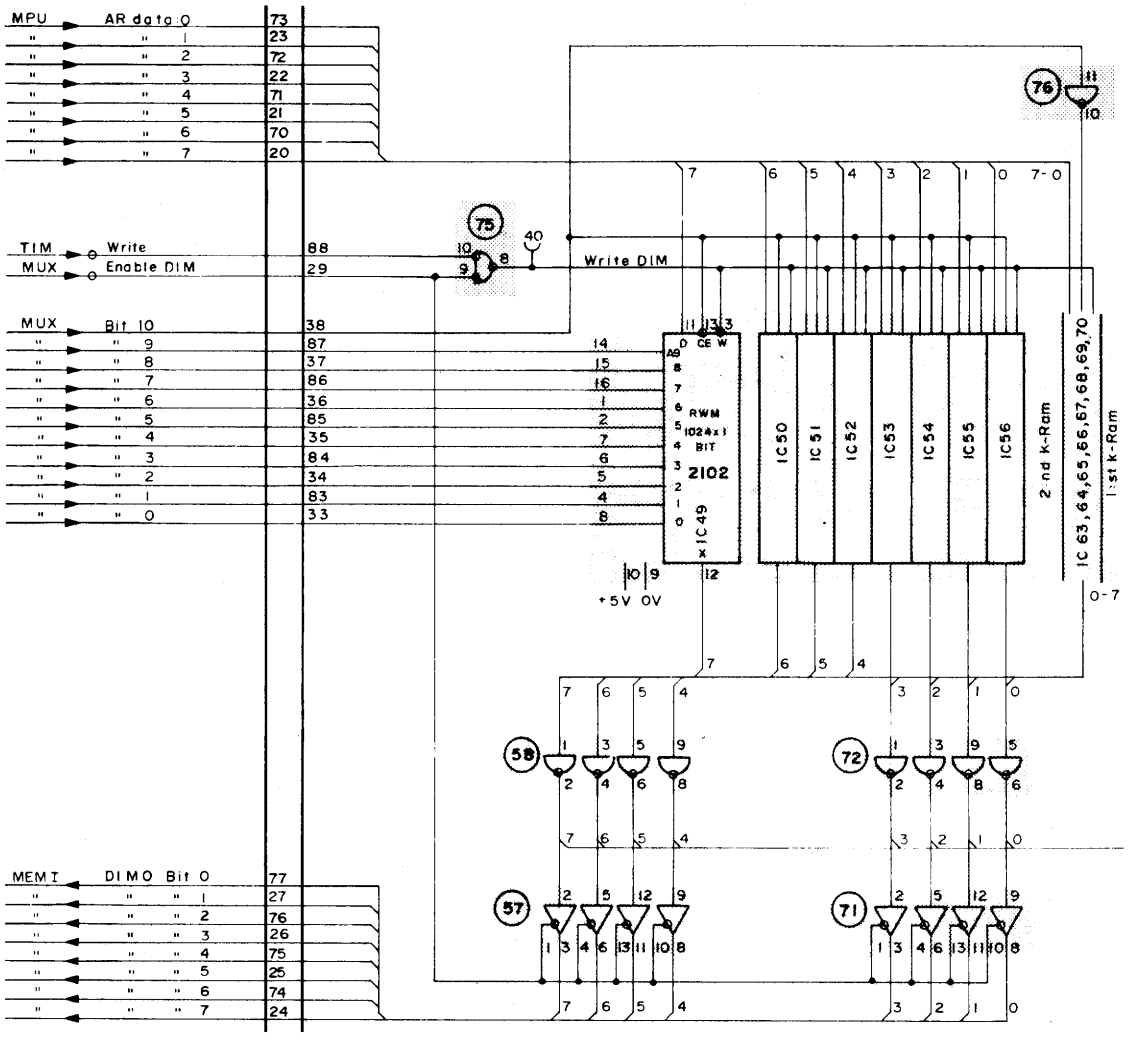


Figure 1. Display memory

Each time DIM is enabled, the tri-state buffer IC57 and 71 is opened and data is fed to DIMO bus. IC58 and 72 feeds low level significant data to line store.

The main part of DIM is thus occupied by characters to be displayed. However, some information of the working memory (WOM) is stored in the DIM, since among the information in the WOM is a part that is exclusively related to a certain display unit e.g. the cursor address and the character which is written under the cursor. The following list is an example of the organization of a DIM in a certain project (RAFA).

Display Memory 0 is used as reference

Byte address	
4000	Display Memory 0
4780	Addr to last RS before cursor
4782	Cursor addr in display memory (2 bytes)
4784	Character under cursor
4785	Indicator for field containing cursor Bit 7 ON – protected field OFF – unprotected field
4786	Send-code from keyboard
4787	Status/sense info + control info (2 bytes) Byte 0 status Bit 7 ON – device lock 6 – edit lock 5 – not used 4 – send request 3 – USM 2 – print request 1 – char not clear 0 – not used
4788	Byte 1 sense Bit 7 ON – device attached 6 – not used 5 – not used 4 – not used 3 – not used 2 – not used 1 – ID failure 0 – ID seq finished
4789	ID-character area (16 bytes) First NUL ends record
4799	Start addr display area
479B	End addr + 1 display area
479D	End addr+1 –1 line disp area
479F	Start addr +1 line disp area
47A1	End addr+1 –2 lines disp area
47A3	Keyboard repetition status

Byte address

47A4	Status for EDIN/EDUT
47A5	Edit module spare
47A6	Counter for alarm time
47A7	Latest order to DIA Bit 7 ON – insert mode lamp on 6 – read ID-code 5 – USM lamp on 4 – alarm 3 – clock 2 – I/O error lamp on 1 – system ready lamp on 0 – retransmit last byte
47A8	IC counter for USM time
47A9	Info at ID-reading

Line store

Line store IC29–30 consists of two dynamic shift registers.

One of the main tasks of the DA is to fetch eight bit character information from the display memory, convert it into readable letters and figures and feed this as video to the screen via STL.

One text line is handled at a time. The text line cycle is controlled from TIM and begins with the reading from the display memory. The memory address is fed from memory address counter in MUX. Memory address counter, which is reset at the beginning of each frame, is clocked at the same rate as the line store during the read phase.

The line store is opened for data entry by the signal DIM to line store. Data is then clocked into the line store by signals from TIM. The read phase corresponds to the time of two sweeps on the screen.

When one line is read into the line store, the input is blocked and the presentation phase starts. The clock pulse rate to the line store is doubled. This implies that the line store is recycled with a rate corresponding to one sweep of the screen.

Character generator

When a key is depressed on the keyboard, eight bits are set in keyboard output register, according to the keyboard code table (an example see figure 1 KB part). From the output register the character is transferred to MPU, where converting from keyboard code to character generator code (see figure 2 CG Code Table) is done.

The eight outputs of the line store are via inverters IC26 and 27 connected to the character generator. The seven least significant bits form the address of the character generator IC37. The address

BIT NO 7 6 5 / 4 3 2 1	CR	0 0 0	0 0 1	0 1 0	0 1 1	1 0 0	1 0 1	1 1 0	1 1 1
		0	1	2	3	4	5	6	7
0 0 0 0	0	SP	@	SP	0	É	P	é	p
0 0 0 1	1	^	|	!	1	A	Q	a	q
0 0 1 0	2	§	\	"	2	B	R	b	r
0 0 1 1	3	⌟	▮	#	3	C	S	c	s
0 1 0 0	4	£	¬	$	4	D	T	d	t
0 1 0 1	5	¢	⋮	%	5	E	U	e	u
0 1 1 0	6	¤	✳	&	6	F	V	f	v
0 1 1 1	7	Ω	;	'	7	G	W	g	w
1 0 0 0	8	Ø	ø	(	8	H	X	h	x
1 0 0 1	9	]	[	)	9	I	Y	i	y
1 0 1 0	10	≡	Ş	*	:	J	Z	j	z
1 0 1 1	11	Æ	æ	+	;	K	Ä	k	ä
1 1 0 0	12	→	▶	,	<	L	Ö	l	ö
1 1 0 1	13	←	◀	-	=	M	Å	m	å
1 1 1 0	14	↑	⌜	.	>	N	Ü	n	ü
1 1 1 1	15	↓	1/2	/	?	O	___	o	¦

Note: MPU converts certain codes from KB to suite CG. Compare with KB code table.

Figure 2. CG Code Table

is altered 80 times each line store cycle according to the stored text line characters. For each character the generator produces a video configuration of seven pulses. The character generator is also controlled by the four bits from the sweep counter in TIM, which is changed one step for each sweep. Thus the generator provides different output each sweep, which is indicated in figure 4. Figure 3 shows the character layout.

Video register

The output of the character generator is connected to the parallel input of the video register, IC38. TIM feeds clock pulses, $\overline{CP}$, to the video register and to the column counter IC28.
The signal, C9, from IC28 is fed to 38/15, because high level on this pin prevent the load pulse to shift data.
The load pulse is formed in 11/12, sheet 5.

Spacing between to positions are necessary (see figure 4). Hence nine clock pulses is fed to the video register for each character (seven dots).

Due to certain short-comings in the display, the trailing edge of the video pulses must be slightly delayed. This is made by using the circuit delay in the phase-shifters, IC23 and IC35.

DA

Column counter

A lot of circuits use the column counter output (C9 see video register). The signal, column 9, is fed to 31/2. Since this signal has low level during a column, the next clock pulse zero-sets FF31/3 and a one is shifted through the register.

The outputs C1, C2, C3, C5, C7, C8 and C9 get positive pulses in an interval of nine clock pulses. C9 each column 9, C8 each column 8 and so on

Field control logic

Certain field control characters can be stored in DIM. Such a character is defined by line store data bits 7 and 6 = 10 (i.e. right part of the eight bit code table) or the underline/blink character 5F. The following cases are used

Underline :5F

	b_3	b_2	
light pen detectable	0	0	normal intensity
	0	1	
	1	0	high intensity
	1	1	non displayed field (blanked)



Figure 3. Character layout

Bits 0, 1, 4 and 5 are used by MPU and are neglected by DA.

Line store data regarding field control are decoded in the decimal decoder IC25 and the two ROM IC15 and 14. The outputs from the decoder are connected to the random logic (sh 5) which provide the following four functions

- Underline or blinking text
- Light-pen detectable field
- Non displayed field (blanked)
- Intensity control (high/normal)

Consider the underline logic. When 5F is decoded at the output of the line store, 1/4 is set by column 8 of the same position. 5F also blocks the setting of the video register in 11/12 via 9/6 and 21/6 in order to prevent the character, which is stored in the character generator at address 5F to be presented at sweep 7; see code table, figure 2.

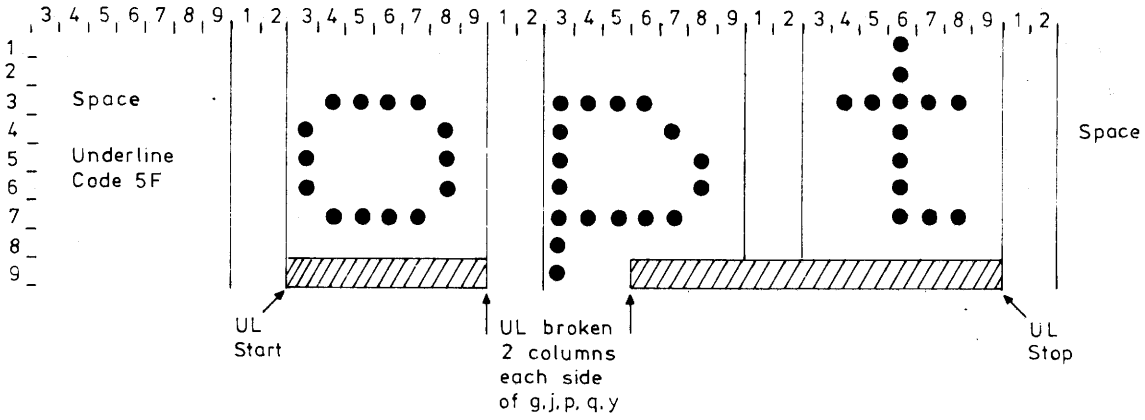


Figure 4. Underline of the word "opt"

When lower case characters with "tails" (descender characters; g, j, p, q and y) or any kind of video appear in the video register during sweep nine, the underline is blocked two columns on each side of the video dots as shown in figure 4.

When underline is valid 40/11, 12 is high and because Format line is low level significant 40/6 is high. If no video signal excists 38/7 and 39/3, 6, 11, 14 feed high level to 40/1, 2, 3, 4, 5. Thereby 40/8 feeds low level to 36/9.
36/8 feeds the video via 7/3, 83/8, 83/12 and 76/8 to the Serial transmission amplifier (sh 5).

If a descender character shall be displayed and a video "dot" appears on sweep nine, UL/LP line is blocked in 40/8 during five columns when the video "dot" is shifted through the Underline inhibit control IC39.

The ONE-side of 1/4 is connected to the J-input of 2/1 via 4/1. At the succeeding column 2, 2/1 is set, which in turn sets the UL/LP line FF22/12 during sweep 9 · display condition (D) · C2.

The strapping plug 19 is in this case strapped 3–2.
It is possible to strap DA to interpret 5F as blinking text instead of UL. In that case 19 is strapped 3–14, whereupon 2/15 is gated with the blink frequency in 7/11.

The initiating FF 1/4 is set only during one position. The underline blink FF 2/1 and also 22/12 are cleared by space or (strappable in plug 19) NUL, brackets or FS, GS (▶ ◀).

Should underline or blink be able to appear over more than one line (i.e. past position 80), 19 is strapped 1–15. This ends the blocking of 5/4, which will be set from 2/15 by sweep 9 in order to preset 2/1 again at the next pos 21 on the succeeding line when presentation shall start.

The functions non displayed field and light pen detectable field has similar functions. The light pen detectable field, however, has no "line wrap-around FF".

The intensity control part contains the intensity FF 3/6, wrap-around FF 5/13, IC18, and 20/12. The working principle of this part is to release 20/13 every time a "high" or a "normal" character is decoded from line store. Thereby 20/12 feeds a low level pulse to the transmission circuits. An Intensity control FF in STL is thereby triggered. This FF is coupled to "divide by two" and it is cleared (= normal) every H-hold. The first low level (0.25 μs) after H-hold is detected as "high" intensity and the second as "normal", the third as "high" a.s.o. See STL section and time diagrams fig. 6 and 7.

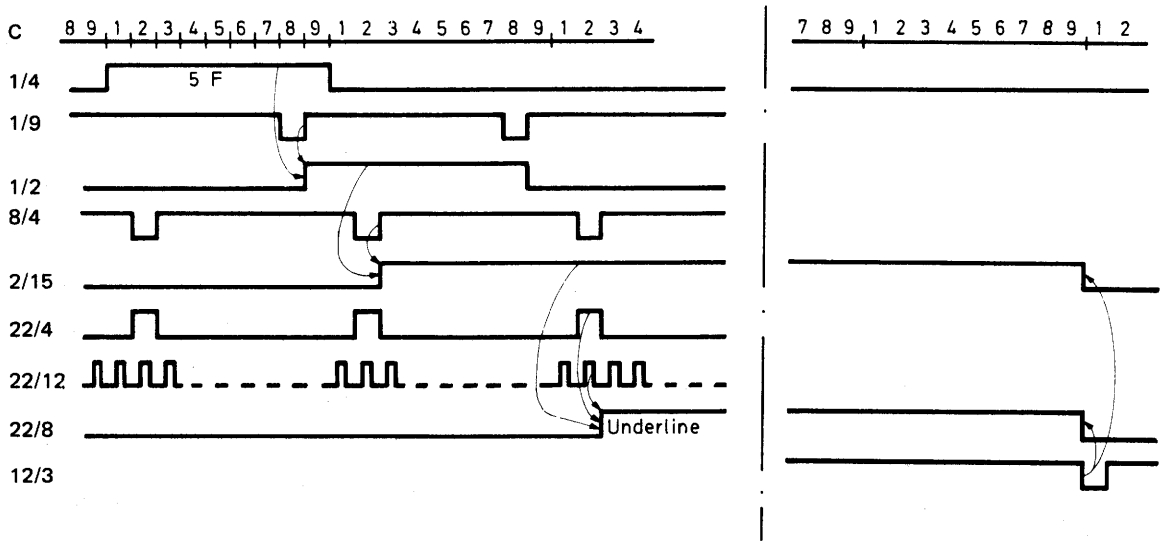
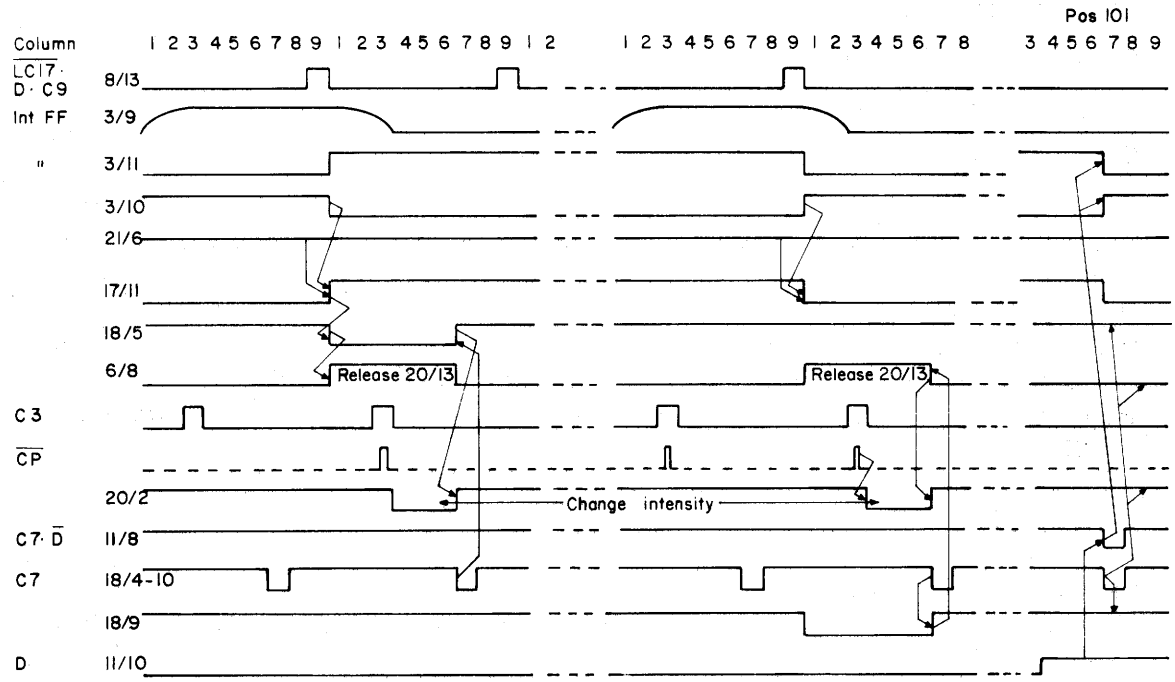


Figure 5. Underline time diagram



High level is reset either at end of line or by the control character "normal" from line store DA

Figure 6. Intensity control time diagram

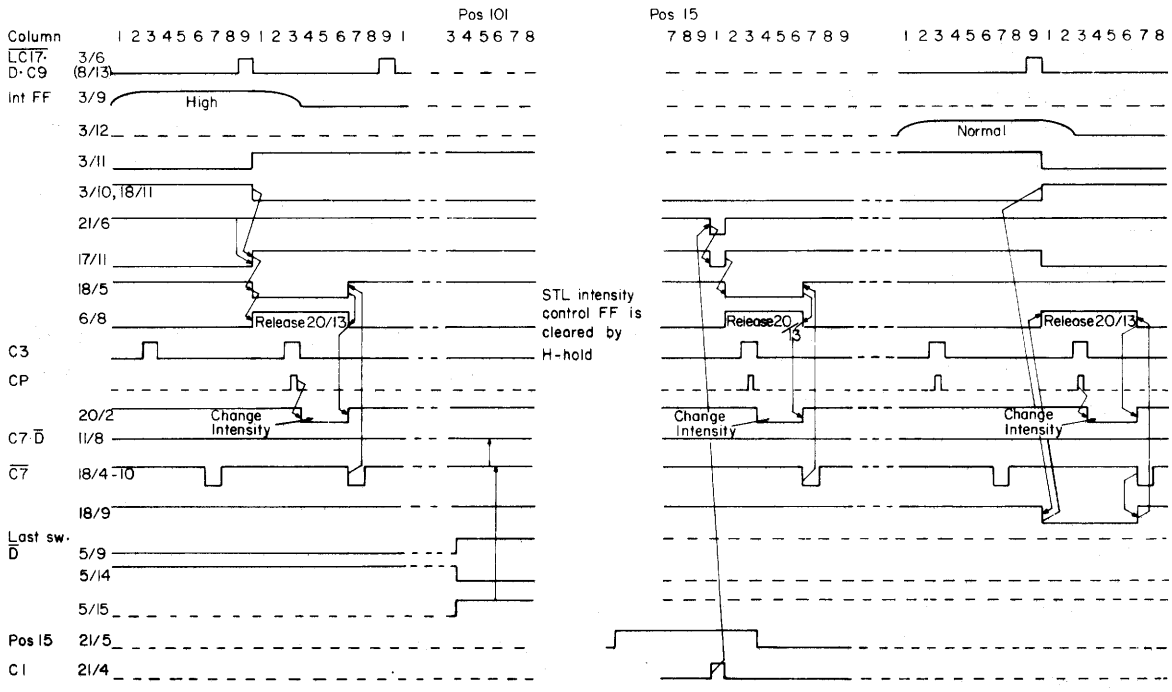


Figure 7. Intensity control with line wrap-around

Data transfer circuits

Also see STL, MUX and TIM sections.

Command byte

The command byte is set into DA command register, IC32, 44, 81 and 45, from MPU via MUX. This may happen any time during a frame and the load pulse contains enable DA and command to display.
The bits retransmit, click and read ID must only appear once each setting and must therefore be treated in a special way. The other command bits are set into IC45. Correct parity is generated in IC33 and the byte is set in parallel into the I/O register by load I/O register from TIM, which appears during the "V" sweep. The byte is then shifted out to the STL by I/O shift pulse containing "H" from TIM.

- Read ID

When a read ID command is applied (low level) from MPU/MUX, 32/3 is set to ZERO. Thereby 44/3 gets a clock pulse and is set. The low level from 44/6 then immediately presets 32/4 (blocks a new read ID). At the succeeding load I/O register leading edge from TIM the (low level significant) read ID command is set into the I/O register. The leading edge of load I/O register also sets the ID message FF 20/9, which in turn immediately clears 44/3 i.e. 44/6 rises to high level which means read ID. This is to prevent read ID from being set a second time the succeeding frame unless MPU has set 32/3 to ZERO again.

The low level from 20/6 is also fed to the status register which is set at the end of byte 1. This implies that 20/6 and the status serves as a memory and a tell-back for MPU regarding read ID, since the status register content is fetched by MPU when MUX rises KB interrupt the next time.

- Retransmit and click

Retransmit and click is treated in the same way in the command register. Consider setting of the click command. The low level click command sets 81/3 to ZERO. Thereby 81/11 is preset which implies that 81/3 is preset (blocked) from 81/8. Thus a new click command cannot be set until 81/8 rises to high level. Furthermore, MPU can set the click any time during a frame and then a new command byte during the same frame without the click bit valid, without knowing if the click has been transmitted or not.

The click-bit (low level) is set by the leading edge of the succeeding load I/O register into the I/O register. The trailing edge of the load-pulse resets 81/11, whereupon the blocking of 81/4 ends and a new click can be set from MPU/MUX.

Tags and status

During byte 1 from TIM the tag byte of five bits is shifted into the I/O register (low level) from the coaxial. The I/O register is thereby clocked by I/O shift pulse, containing pos 53.
This byte contains:
LP activated
Repeat
Strobe
LP hit
Odd parity

- LP activated means that the light pen has detected a lit video dot and thus activated the light pen logic. When the bit is valid, the light pen activated FF 60/11 is set to ZERO by load status register from TIM. Thereby the field control circuits concerning light pen detectable fields are activated.

- Repeat means that a move cursor key or a tabulator key has been depressed on the keyboard for more than 0.5 seconds. This tag is set into the status register by load status register from TIM.

- Strobe is fed to the OR-gate 36/6 to prevent the NUL message FF to be set to ZERO; see below in paragraph Interrupt and overrun.

- LP hit indicates that the light pen has detected an activated field and a counter in the light pen logic holds the number of this field.
  Thus the succeeding data in byte 2 should be considered as light pen data and therefore this bit is set into the status register to notify MPU.

- The tag byte is parity checked in 61/9 and if a parity error is detected, the data register IC47, 48 and 31/11 is cleared. MPU thereby fetches NUL with wrong parity.

Interrupt and overrun

If no status bit or strobe is valid (ID, rep, strobe or LP hit), it is obvious that DA interrupt must not be fed to MUX. In this case the NUL message FF 82/3 is set and thereby setting of the interrupt FF 61/12 is blocked in 83/5. If, however, a status bit or strobe is valid, 82/3 is set to ZERO and thus the interrupt FF is set by byte 2 · load data. Thereby the low level interrupt is fed to MUX via 34/8, which is opened by enable DA.

The interrupt FF is cleared by enable DA · data from KB (INKB). If MPU/MUX has not fetched a previously set data, the interrupt FF is not cleared and when a second set pulse is applied, the overrun FF is preset from 59/3. The overrun status is then fetched by the leading edge of DA status to MUX. The trailing edge of DA status to MUX resets the overrun FF, which in turn provides clear condition for the interrupt FF in 41/4.

Data byte

The data byte (byte 2) is shifted into the I/O register by I/O shift pulse (pos 53) from TIM. The byte is set into the data register when the interrupt FF is set by load data from TIM. Byte 2 is parity checked by MPU.

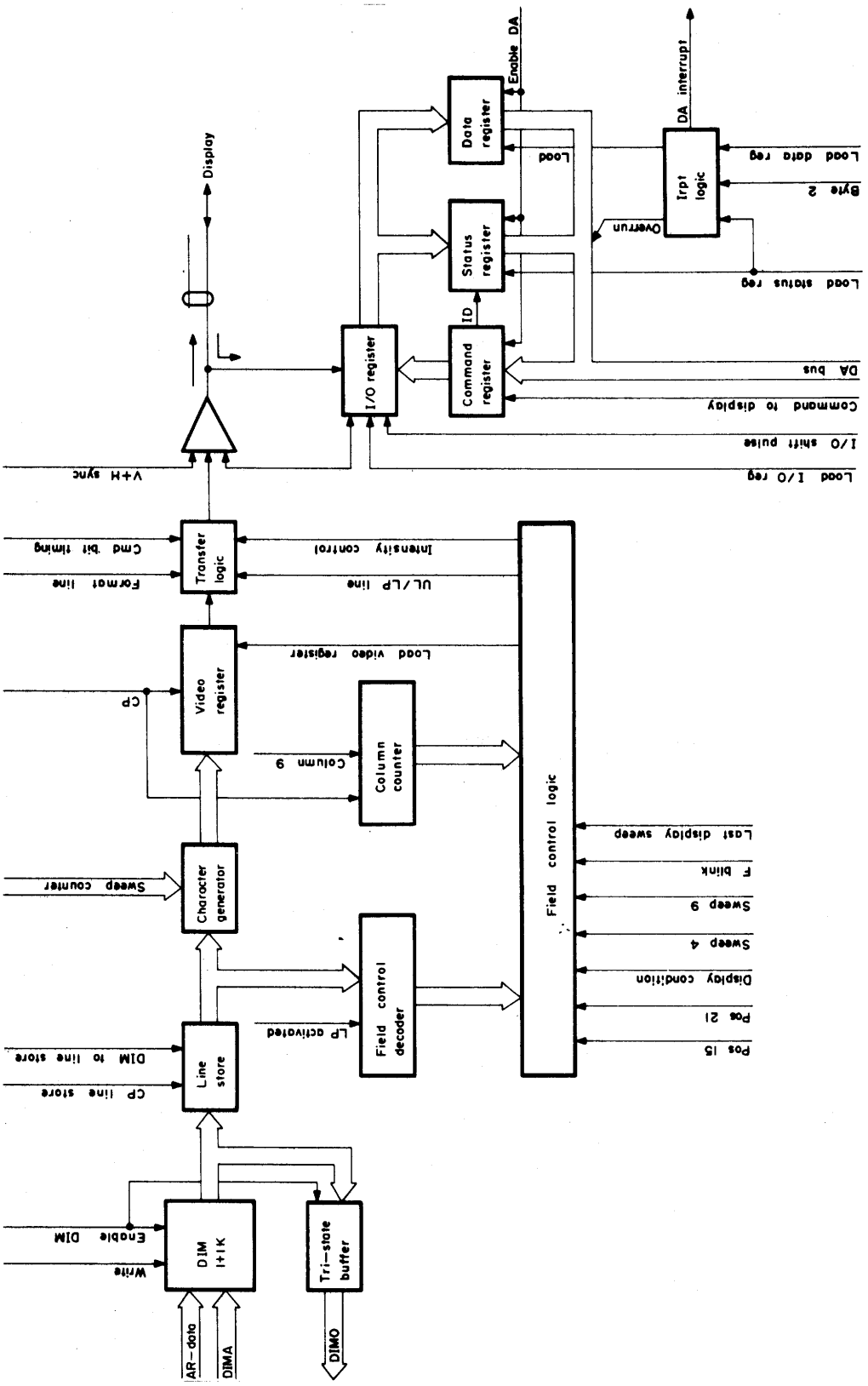


Fig. 8. DA block diagram

Printer Adapter 3533

Contents

BRIEF OUTLINE 1

- Printer memory 1
- Start of printer 1
- Character to printer 1
- Stopping the printer 3
- Parity error 3

DETAILED DESCRIPTION 3

- Request to send to PR 3
- Delay of PA interrupt 4
- Load 4
- Delaying the turning-off of the printer motor 4
- Parity error 4

FIGURE

1. PA block diagram 2

Printer Adapter 3533 comprises one PC-board of the small size (half board). It is used in Control Unit 3501/3502 if a Data Printer 3571 or 3572 is connected. Straps can be connected to provide one of three transmission rates. MUX provides clock signals suitable for three of the following rates: 9 600, 4 800, 2 400, 1 200, 300 and 109 bit/s. Moreover, it is possible to select between one and two stop bits and between odd and even parity by means of strapping. Strapping is also used to determine whether or not the reversed channel from the printer adapter can stop transmission.

BRIEF OUTLINE

The block diagram in Fig. 1 shows the printer adapter operation.

Printer memory

The PRM comprises 2K RWM IC26 – IC41. After a PRINT key has been depressed, display information is transferred from the display memory to PRM. See also the section entitled "Display memory" in the display adapter description.

Start of printer

The start word, 10xxxxx (bits 7–0) is transferred from MPU to PA via MUX and the PA bus. The motor control circuits issue the Request to send to PR signal which starts the printer motor.

Bits 3–0 are set in the timer which determines the delay before the PA interrupt is transferred to MUX. This delay is necessary since it takes a certain amount of time for the motor to start.

Character to printer

Character information sent to the printer always contains bit 7 = 0. Bits 6–0 present the character.

When MUX has enabled one PA (which provides a PA interrupt) MUX sends the PA status to MUX signal to PA (and sends Printer interrupt to MPU). The status obtained from the printer consists of the Request to send, Parity fault and Data terminal ready signals. The PA status to MUX signal opens a tri-state buffer which sends the status to MUX. MPU fetches the status word from MUX.

Then MPU fetches the character from the printer memory and transfers it via MUX and the PA bus to the terminal transmitter. The character is entered into the transmitter by means of the Data to printer signal.

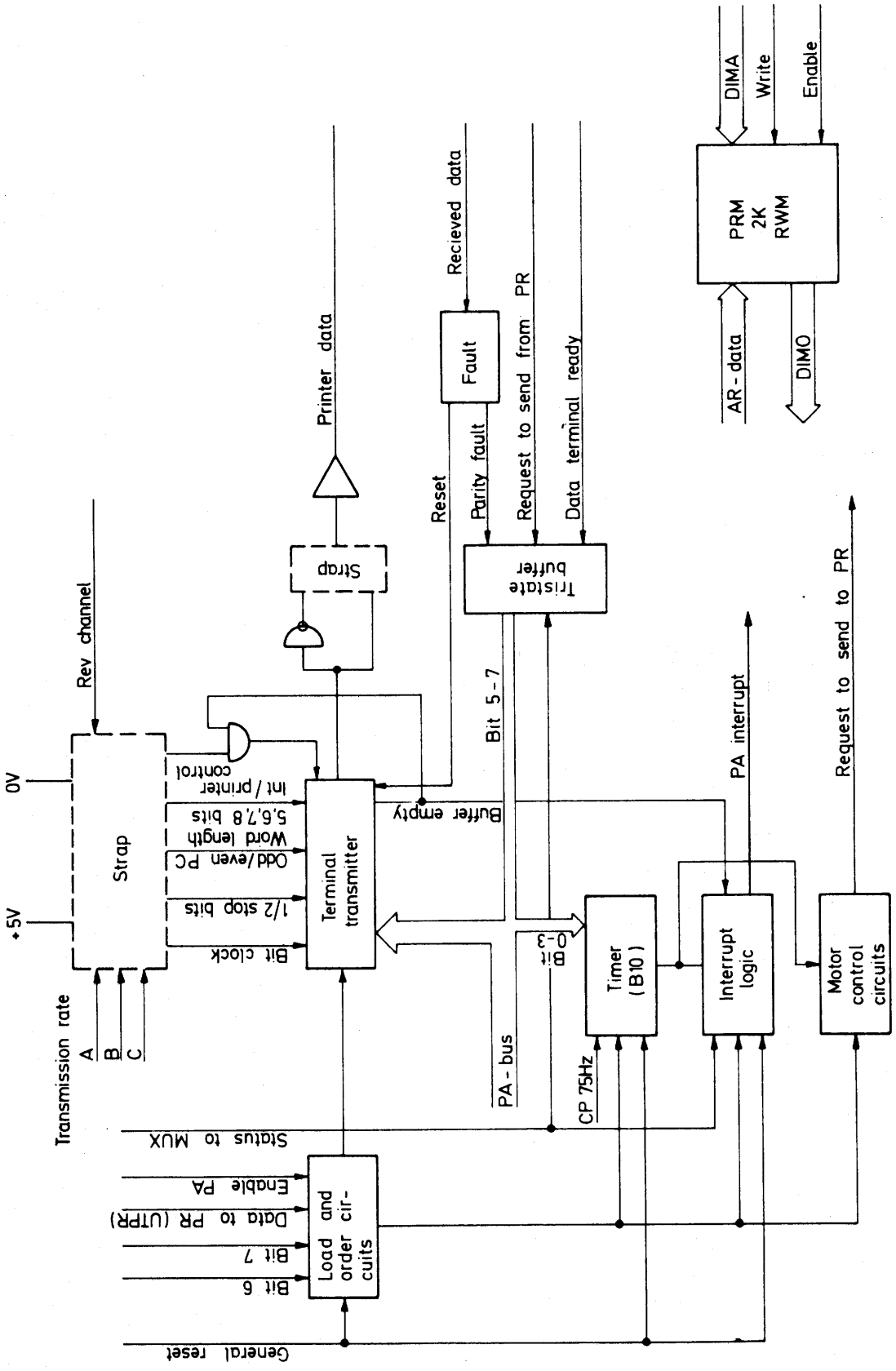


Fig. 1. PA block diagram

The terminal transmitter is an LSI circuit. The transmitter sends data at a speed determined by the selected signal obtained from MUX. Start and stop bits as well as parity check bits are generated in the transmitter. The polarity of the output signal can be selected in a strapping plug.

The terminal transmitter has one shift register (for sending) and one buffer register. When one character has been sent, the next character is transferred from the buffer register to the shift register for subsequent sending to the printer. Simultaneously, the buffer empty signal is sent to the interrupt logic and to the transmitter control gate. If the printer adapter is strapped for internal control (or if the printer is ready to accept a new character if strapped for printer control), the sending of the next character is started.

The interrupt logic issues the PA interrupt signal to MUX and MUX issues an interrupt to the MPU. Thereafter, the MPU has at its disposal one character transmission time, within which it must send another character to the transmitter (if transmission shall be continuously).

Stopping the printer

The stop word 11xxxxx (bits 7–0) is transferred from MPU via MUX and the PA bus. The printer must have time to print the last characters before the motor stops. Consequently, the turning-off of the motor is delayed. This is accomplished by setting bits 3–0 in the delay counter (in the timer) which is decremented until it provides borrow, thus causing the printer motor to be turned off.

Delaying printout when the printer is turned on is accomplished by means of start word 10xxxxx, in which bits 3–0 determine the delay of the PA interrupt.

PA

Parity error

The printer checks parity and, if it detects an error, it sends a pulse (received data) to PA at the end of the character. This character, as well as the next character which is normally already in the transmitter buffer register, must be sent out again from the MPU.

The pulse resets the transmitter, whereupon the stop bit is "prolonged" while PA simultaneously sends the PA interrupt signal to MUX. When MPU receives the status word, it senses the parity error and sends out the character that was transmitted erroneously.

If the transmitter buffer is empty, PA waits for the character from the MPU, before sending the status word containing the parity error. Consequently, MPU has always sent out one character after the erroneous character.

DETAILED DESCRIPTION

Request to send to PR

The signals Enable PA and Data to printer, as well as start word 10xxxxx (bits 7–0) provide a clock pulse at 9/11 via 4/10. 9/11 is reset since 9/12 is at the low level (bit 6 = 0). 9/9 presets 8/11. 8/8 sends Request to send to PR via 16/3. The motor starts.

Delay of PA interrupt

Counter IC5 is set by means of bits 0–3 in the start word. The setting pulse is obtained from 4/10 via 4/1.

The CP delay of 75 Hz, obtained from MUX, is divided by 10 in IC2. This square-wave signal, having a frequency of 7.5 Hz, decrements counter IC5 until it provides borrow. When this occurs, the counting pulses are blocked by 4/4.

Examples of delay times:

If bits 0–3 are 0000, there is no delay.
If bits 0–3 are 1000, (bit 0 = 1) the delay will be approximately 130 ms.
If bits 0–3 are 1111, the delay will be a maximum of about 2.1 s.

4/13 provides the high level at 7/2.
7/1 is at the high level since 9/11 is zero-set.
The buffer register in the transmitter is empty. The transmitter then provides the signal Buffer empty (20/20) which is sent to 7/13 where it provides the high level.
This means that when IC5 provides borrow, a clock pulse is obtained at 8/3. 8/3 is zero-set and 8/6 provides a PA interrupt via 13/13. The tri-state buffer 14/3 permits the interrupt to proceed to MUX, when MUX enables PA.

Load

When a character is to be entered into the transmitter buffer register, 20/21 shall have the high level.

Signals Enable PA and Data to printer are sent to 7/11 and 7/10 respectively. If bit 7 is simultaneously zero, 11/8 provides the high level at 20/21 via 17/8.

Delaying the turning-off of the printer motor

When the printer has printed everything that is to be printed, MPU sends a stop word 11xxxxx (bits 7–0) to PA via MUX. No new character is entered into the transmitter buffer register. Signals Enable PA, Data to printer and bit 7 provide a clock pulse at 9/11 and a setting pulse at IC5. 9/11 is set since bit 6 = one. Bits 0–3 are entered into counter IC5 and determine the delay time in the same way as used for the PA interrupt delay. When counter IC5 provides borrow, 8/11 is cleared by 7/6. 8/8 receives the high level and the printer motor is turned off. 8/9 goes low, which provides the high level at 20/7 and 20/9. The high level at 20/7 and 20/9 resets the transmitter.

Parity error

In the event of a parity error, a positive pulse (Received data) arrives which commences during the PC bit and endures for one bit time. 9/3 is zero-set by the next CP. 9/5 goes low and clears 6/1. 24/6 indicates a parity error after signals Enable PA and Data to printer have zero-set 6/6. When 5/15 goes low, the transmitter is reset. The buffer empty signal provides a clock pulse at 8/3, while 6/14 is at the high level so that 8/3 will remain set. 10/6 is at the high level and 9/5 obtains the high level after the next CP, thus causing 11/11 to go low.

Situation 1: One character in transmitter buffer

6/11 is low after the character has been entered into the buffer. Since 8/3 has not been zero-set, 13/4 provides a PA interrupt via 13/13 and 14/3.

Situation 2: No character in buffer

If the transmitter does not contain any character when the printer signals a parity error, signal PA status, which is sent to MUX, must set 8/3, and signals Enable PA and Data to printer must zero-set 6/6 before 24/6 indicates a parity error. That is to say, one additional character must be transmitted before 24/6 indicates a parity error. This character cannot be entered into the transmitter buffer register since 20/7 and 20/9 are at the high level. 13/4 provides a PA interrupt via 13/13 and 14/3.

MUX senses the PA interrupt and sends the PA status to MUX signal to PA. Bit 6 in the status word specifies that a parity error is present, and MPU re-transmits the two most recently transmitted characters.

The signal status to MUX clocks 6/1 via 1/8. Thereby 6/1 is set. 20/7 and 20/9 go low and the transmitter buffer register can be loaded.

Micro Processing Unit

Contents

GENERAL 1

FUNCTIONAL DESCRIPTION 3
- Instructions 3
- Microinstructions 4
- Microprogram 8
- Interrupts 12
- Timing 13
- MPU relations to other units 14

DETAILED DESCRIPTION 15
- Instruction register 15
- Selectors 15
- Selector 1 15
- Microinstruction counter (MIC) 15
- Type decodes and latches 16
- Timing circuits 17
- Interrupt circuits 17
- Condition flip-flop circuitry 18
- ALU 20
- AR 21
- Stack and register 22
- B-selector 23
- Mie decoder 24
- Instruction counter (IC) 24

FIGURES

1. Block Diagram of MPU 2
2. MPU timing 13
3. Carry FF and zero FF 19
4. Shifting conditions 21

APPENDICES

1. Microinstructions
2. Control ROM
3. Microprogram

GENERAL

The micro processing unit (MPU) comprises two PC-boards: MPU1 (E07150 0000) and MPU2 (E07150 0010). The MPU is an essential part of all controllers in the 3500 family: Control Unit 3501, Control Unit 3502, Display Station 3550 and Control Unit 3170 A. It controls the operation of the controller, governed by instructions stored in the memory (MEM).

The MPU handles the instructions according to a microprogram memory, that governs the controlling part of the MPU. The microprogram is fixed, i.e. is affected neither of which controller the MPU is incorporated into nor of the application.

Before reading the following description of the MPU, the reader should be familiar with the principles of controller operation. Please refer to sections entitled "Brief outline" in the 3501 part and "Software information" in the part "General concepts". The latter section also contains a simplified block diagram of the MPU.

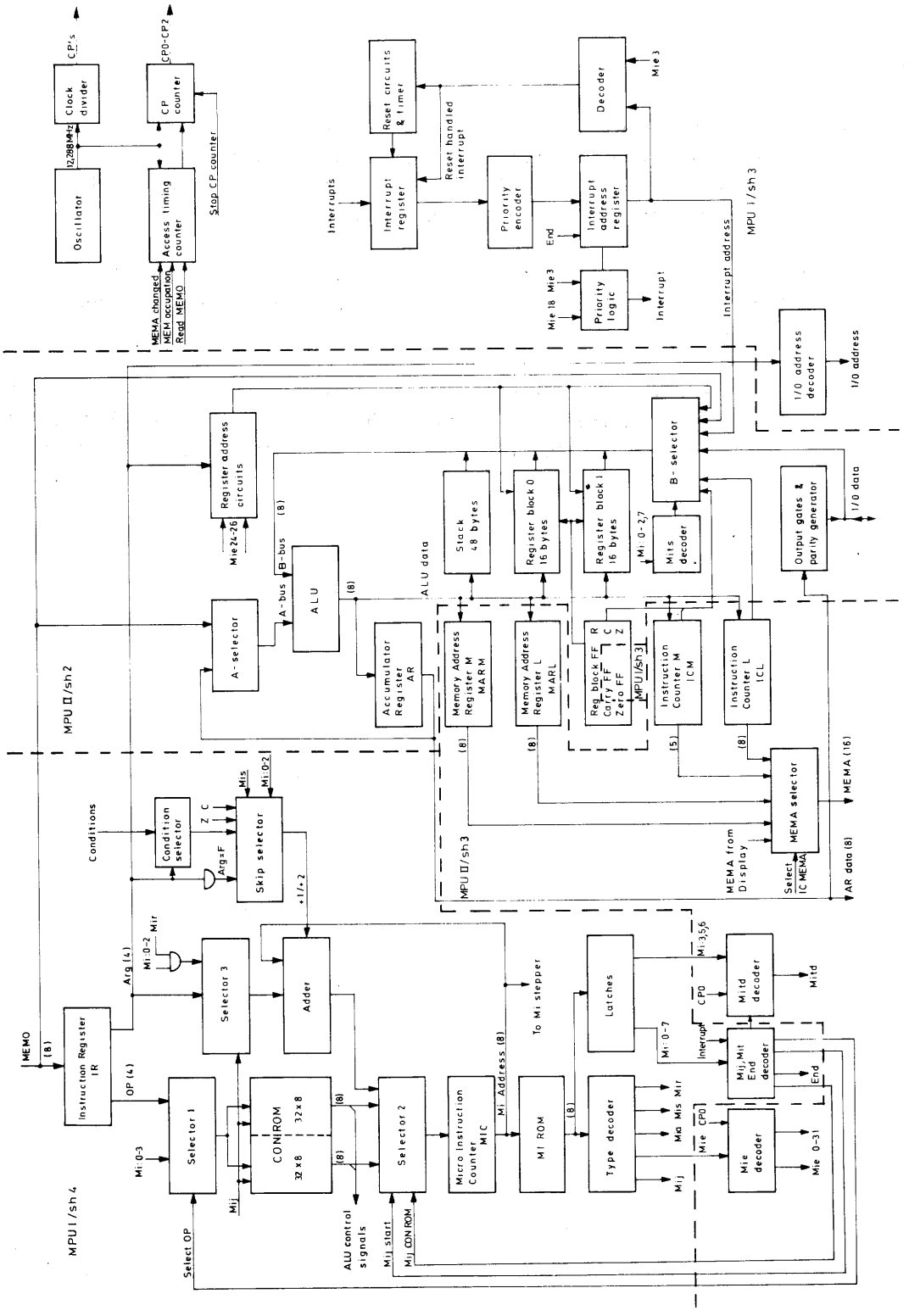


Fig. 1. Block Diagram of MPU

FUNCTIONAL DESCRIPTION

The following functional description is mainly based on the block diagram in Fig. 1. This block diagram is similar to the simplified block diagram in section entitled "Instructions", but it contains more blocks. The block "Control logic" of the simplified diagram is split up into the blocks in the left part of Fig. 1. The numbers within parenthesis states the number of bits on the busses etc.

Instructions

The instructions that arrive from the memory (MEM) call for several operations within the MPU. MPU operations are controlled by microinstructions. To make the operations specified by the instructions, a minimum of four microinstructions are needed. Complicated instructions require more microinstructions (up to 30 microinstructions for the most complicated instructions).

The microinstructions are stored in a read-only memory (MI ROM) of high speed bipolar type. Generally speaking, the microinstructions belonging to a certain instruction are stored in consecutive cells, but in case a sequence of microinstructions suits several instructions, the sequence is common to the instructions in question. The last microinstruction in an instruction containe a mark bit (end bit) which zero-sets the microinstruction counter (MIC) that addresses the MI ROM.

During execution of a byte in an instruction, the instruction counter (ICL and ICM, 8 least significant bits and 5 most significant bits respectively) holds the address to the next memory cell. This means that the next byte is available at the memory output (MEMO). The byte under execution is stored in the instruction register (IR).

When an instruction is completely executed, the microinstruction counter is zero-set by the end bit in the last microinstruction. The microinstruction found in the first cell of the MI ROM is decoded and causes a) the first byte of the next instruction to be entered into the instruction register and b) the instruction counter to be incremented by one (thus, fetching of next byte starts immediately).

The microinstruction counter is subsequently stepped to 1 and in the second cell another microinstruction is found. That microinstruction gives raise to signals Mij (microinstruction type j) Select OP and Mij start. Then, the OP part of the first byte (the four most significant bits) is fed to CONROM (control ROM) via selector 1. The CONROM is composed of two halves each containing 32 bytes with 8 bits. As signal Mij is present, one of the 16 most significant bytes of CONROM (in both halves) is addressed (see Appendix 2, page 2). Consequently, CONROM delivers two 8-bit bytes to selector 2. Signal Mij start selects the byte arriving from the left part of CONROM (the start address part). The microinstruction counter is thus loaded with a new address that has been determined by the OP of current instruction.

The third cell that is addressed contains the first microinstruction that carries out an operation called for by the instruction. The two first microinstructions are common to all instryctions. The only thing they do is to get the address to the cell where the first "working" microinstruction of an instruction is stored. In this and subsequent cells, the microinstructions that see to it that the instruction is executed properly are stored.

Microinstructions

A summary of all different microinstructions is presented in appendix 1. Note that the summary present all possible microinstructions. The existing microprogram does not use all these microinstructions.

The microinstructions contains eight bits, but one to five bits are needed for classification and in three types of microinstructions one bit is needed as end bit. That leaves a maximum of six bits that can specify an operation.

A microinstruction can be specified in three ways

- the type plus a reference number (decimal), example Mie 17
- the eight bits in the microinstruction (Mi bits), same example 0110 0001
- the hexadecimal number given by the eight bits, same example 61

The first of these three ways gives no information about the end bit.

Microinstructions type t

The microinstructions that are used for transferring information (one byte) from one location to another are called t-type. The six "usable" bits are divided in two groups, each with three bits. This is made to facilitate decoding of eight sources of information and eight destinations for the information. Totally 64 t-type microinstructions are thus available, but its evident that e.g. a transfer from ICL to ICL is equal to no operation (No op).

In the microprogram only 14 different t-type microinstructions are used. By way of example, the byte from I/O data is only transferred to AR, C and Z, that in turn only gets information from I/O data.

When a t-type microinstruction is executed, control signals are sent from CONROM to ALU. (See Appendix 2.) The ALU operation is to put eight-bit bytes from the B-bus to the ALU data bus.

The three least significant Mi-bits are sent to the Mits decoder along with the most significant Mi-bit. The Mits decoder makes the B-selector select the source of information. Of course, the Mits decoder also controls the stack and register so that data are not mixed on the B-bus.

The three Mi-bits that specifies the destination are feed to Mitd decoder. This decoder sends set-pulses (at proper time) to the ICL, ICM, etc.

Microinstructions type e

A somewhat heterogeneous group of microinstructions are called type e (execute). The name is realy covering for only six of the microinstructions. These six are the actual "executing" microinstruction of the instruction (example Mie 16 – disable interrupt). The other e-type microinstructions give raise to signals that controls IR and to pulses that set flip-flops, increment counters or the like. Several of these microinstructions can be needed to fully execute the instructions. Some complicated e-type microinstructions are discussed in the following.

Mie 0 and Mie 8

In the instruction register (IR), the bytes are stored in the following way: OP:3 (MSB), Arg:3, OP:2, Arg:2, OP:1, Arg:1, OP:0, Arg:0. If the bits are shifted one step to the left the argument will be moved to OP position. This is done by Mie 0. Microinstruction Mie 8 moves the OP to argument position by shifting right. When shifting left the MSB (most significant bit) is entered as LSB (least significant bit) and when shifting right the LSB is entered as MSB. Consequently, if IR is first shifted right and then left, the original positions of the bits are resumed.

Mie 3

When an interrupt is to be executed, some microinstructions are issued before the first instruction of the interrupt program. Mie 3 is included among these microinstructions. It sets the priority of the interrupt program, resets the interrupt flip-flop (to make it possible to store another interrupt signal immediately). Moreover, it should zero-set the most significant part of instruction counter (ICM).

However, the counters are built up by four bit counters that the manufacturer has intended for use in positive logic systems. The outputs from IC are considered negative logic ("one" = low). Thus the counter is set to F (1111) when a pulse is applied to its "zero-setting" input. If another microinstruction (Mie 13, increment ICM) is issued afterwords, the counter will be zero-set. (Note that the input assigned count down by the manufacturer must be used to count up.)

Mie 4

The most significant part of IC is composed of a four bit counter and a flip-flop for the most significant bit (IC:12, also called extention bit). This flip-flop cannot be affected directly, instead, microinstruction Mie 4 sets an auxiliary flip-flop. The auxiliary flip-flop is reset when a microinstruction of t-type with destination ICM is issued. That t-microinstruction sets IC:12 to the new value. As long as the auxiliary flip-flop is set, all interrupts are inhibited. This means that the instruction containing Mie 4 (4B, EXTN - change extension bit) should be followed by CALL or JUMP (that contains the requisite t-microinstruction) immediately.

Mie 7

The stack counter is automatically incremented after a byte is loaded into the stacks. To decrement the stack counter a separate microinstruction, Mie 7, is needed.

Mie 14

The logic and arithmetic instructions specify, in general, an operation where the two operands are the contents of AR and the contents of a register (specified by the argument). If the argument is F, the second operand shall be the next byte instead of the register contents. In the microprogram a test is made to asertain whether the argument is F. If so, the microinstruction Mie 14 is issued. Then a flip-flop (MEM to B Mia) is set, causing B-selector feed MEMO to B-bus in following microinstructions type a (which are the executing microinstructions of logic and arithmetic instructions).

Mie 24, 25 and 26

When a register pair is specified in the instruction, the argument should give the address to the register containing the most significant byte. During execution of the instruction, it must be possible to carry out operations on both registers. The microinstructions 25 and 26 are used in the microprogram to select one of the registers in the register pair.

In the instruction with OP part 5, the MSB of the argument specifies whether it is a load or a store instruction. The MSB must thus be zero-set (by Mie 24) before the argument can be used as register address.

Microinstructions type a

The ALU operations (and some AR operations) are specified by a microinstruction type a. The ALU requires six control signals (S0–S3, MC and Carry in) to operate properly. Some other circuits also require control signals in order to fulfil the operation specified by the microinstruction. These control signals are received from the CONROM. See Appendix 2, page 1. The address to CONROM is derived either from microinstruction bits or from the OP-part.

Data pass the ALU in all a-microinstructions except Mia 2 and 3 (AR shifts). Any calculation is made in the ALU. The A-selector is in position AR data for all microinstructions. The only exception is Mia 4, when A-selector selects MEMO to A-bus.

Microinstructions of a-type with numbers 8–14 are modified if flip-flop MEM to B Mia is set, see section entitled "Microinstructions type e, Mie 14".

The data transfer specified in Mia 8 can also be achieved by using a t-type microinstruction. However, Mia 8 also affects zero flip-flop (which is not affected by Mit).

If the four "specifying" bits (A, B, C and D) in the microinstruction are all ones, the address to CONROM is derived from the OP part of current instruction. This modification makes it possible to condense the microprogram (see Appendix 3, page 1) so that all instructions with OP part 8–E can use the same sequence of microinstruction. As a matter of fact, Mia 9–13 are only used in this modified way.

Instruction 4A (ARIC) also comprises a modified a-microinstruction (A through D ones). As the OP part is 4, the operation of ALU will be addition of A-bus and B-bus. ICL is fed to the B-bus (as in unmodified microinstruction) but the A-selector is set to select AR data (which is the common setting of the A-selector).

Microinstructions type j

A jump in the microprogram is made if the microinstruction counter (MIC) is loaded by a number (that is not the old number plus one). There are three microinstructions that causes a jump (Mij, Mis and Mir). The microinstructions of j-type can be subdivided into three groups as can be seen in Appendix 1, page 4.

Mij (1F)

As already mentioned in the section entitled "Instructions", page 3, the second microinstruction in all instructions is Mij (1F). When this microinstruction is found:

a) signal Select OP sets selector 1 to feed the OP part to address inputs of CONROM,

b) signal Mij gives a one in the MSB to CONROM and

c) signals Mij start and Mij CONROM set selector 2 to feed the start address from the "left" part of CONROM (see Appendix 2, page 2) to MIC.

If the priority logic issues the signal interrupt, signal Mij start is inhibited (and signal Mij start dis is issued instead). Then, selector 2 selects the address (old address plus one) from the adder.

Mij (1E)

If microinstruction 1E is found, selector 3 feeds the argument to the adder. Then the adder adds

a) argument (in inverted form)

b) previous address from MIC

c) one (from skip selector)

which means that a jump, according to the argument, is made in the microprogram (1 to 16 bytes forward). Selector 2 is set to feed data from the adder to MIC (by signals Mij start and Mij CONROM).

Other Mij

If the four "usable" bits of a j-type microinstruction have any hexadecimal number 0 through D, a "microinstruction controlled" jump is made.

a) Selector 1 selects Mi:0–3 as address LSB to CONROM (in inverted form)

b) signal Mij gives a one as MSB to CONROM and

c) selector 2 is set to feed a "label" address from the "right" part of CONROM (see Appendix 2, page 4) to MIC.

Microinstructions type s

Whenever the microprogram shall continue along different paths, depending on some condition, a microinstruction type s is used. The skip selector can give either signal +1 or signal +2 to the adder. For all other microinstructions but s, it gives +1. For a microinstruction type s, the three "usable" bits selects a condition (see Appendix 2, page 4). If the selected condition is true, the skip selector gives +2, otherwise +1.

A microinstruction type s is used to check the condition of a relative jump instruction. The argument, which specifies the condition to be tested, controls the condition selector. The selected condition can be inverted or not (as selected by the s-microinstruction) before being tested in the skip selector.

Microinstruction type r

When a microinstruction type r is encountered, the gate above selector 2 is opened. The three "usable" bits of the microinstruction is thus fed (in inverted form) to the adder. The adder then adds

a) four times the number specified by the microinstruction bits (in inverted form) since these bits are fed to the adder two positions displaced from the least significant position,

b) previous address from MIC and

c) one (from skip selector)

Microprogram

The complete microprogram is listed in Appendix 3. The list contains the address to MIC and the microinstructions specified in three ways. Moreover it contains labels and an abbreviated "description". The labels make it possible to follow the jumps without using Appendix 2 (start addresses and label addresses of CONROM) or without calculating (relative jumps, Mir). In the "operation" column for microinstructions Mij and Mir the label is written within parenthesis. The same label is found at the address where the microprogram continues.

If any microinstruction of types t, e or a is executed, selector 2 feeds the adder outputs to MIC. When the microinstruction is almost completed, the address from the adder (old address plus one from skip selector) is entered into MIC (if end bit is not set). If any microinstruction of types j, s or r is executed, the result will only be a new value in MIC.

The first two microinstruction is common to all instructions. These two instructions are inhibited (become no op) if an interrupt shall be made. After these two microinstructions have been executed, MIC holds an address that is determined by the OP part of current instruction. The OP part at hand can be found in the label column.

Below follows a discussion of some aspects of how the microprogram is composed in connection with some instruction as examples.

Instructions having OP parts 0 and 1

In both jumps and call instructions, the instruction counter shall be loaded by a value given in the instruction. Therefore the two last microinstructions are common to these two instructions.

If the OP part is 1 (CALL), the address in MIC will be 9 after the first two microinstructions. The microinstructions at addresses 9 and A put the old value of IC along with register block selector flip-flop (R), carry flip-flop (C) and zero flip-flop (Z) into the stack. Then, at address B, the second byte of the instruction is transferred to ICL (8 least significant bits of instruction counter) whereupon the argument is transferred to ICM.

If the OP part is 0 (JUMP), the microprogram will proceed to address B after the two first instructions. In this case new data from the memory (MEMO) have not arrived after the memory address (MEMA) was changed. However, each time MEMA is changed, an access timing counter is started. If an attempt is made to use the memory output (MEMO) before the access timing counter has

timed out, the clock pulse generator of the MPU is stopped. Thus the MPU operation is stopped and will not continue until the MEMO information is correct.

Instructions having OP part 2

A skip microinstruction is found as the third microinstruction at address D. This microinstruction feeds the selected condition to the skip selector. If the condition is false, the adder adds one to the old address i.e. next microinstruction will be taken from address E. Thus the instruction counter will be incremented which means that the second byte of the instruction will be dispensed with.

If the condition is true, next microinstruction will be taken from address F (old address plus two). Then the second byte of the instruction will be used to calculate a new value for the instruction counter.

Instructions having OP part 3

Operation for instructions having first byte 30 through 3E is analogous to the above described. But since the operation for 3F differs, a check must first be made to ascertain whether the argument in 0–E or F.

The third microinstruction is found at address EF. If argument is F, the microprogram continues at address F1 by saving old IC value and subsequently calculating a new IC value. For all other arguments a jump is made to address 10 (label JCF).

Output and input instructions

The microprograms for output (OP 6 at address E3) and input (OP 7 at address D6) instructions have only minor differences. These instructions comprise one byte, but if the argument is F they are extended to two byte instructions. The argument is decoded to I/O address sent to the I/O address bus (if arg is 0–E).

In the microprogram, a check is made to ascertain whether the argument if F by using two skip-microinstructions. Only if the argument is F, the microinstruction 41 (Mie 1) shall be executed, for all other arguments if must be skipped. There exists no single microinstruction to make this, but by using the unconditional skip 0E (note that Mark (carry) is reset by the first microinstruction at address 0 and is thus = 0) as the second skip instruction, the result will be the same.

Before the microinstruction E8 (I/O data → AR) or 67 (reset I/O data from MPU) a no op microinstruction is executed. This is made to give the "peripheral" unit ample time to put data to (or read data from) the I/O data bus. The no op after the last "working" microinstruction ensures that the data on I/O data bus are valid well beyond it is used.

Instructions having OP part 8–E

All logic and arithmetic instructions as well as the load instruction have the microprogram in common. The different operations are achieved by giving different control signals to the ALU. The Mia that in used causes the OP part to address the CONROM and thereby determine ALU operation.

The argument selects which register puts data to the B-bus (if it is 0–E), while the A-selector feeds AR data to the A-bus. If the argument is F, the A-selector is changed to feed MEMO to the A-bus. Thus, the instruction becomes a two byte instruction which means that IC must be stepped after the operation is carried out. Then the MPU is stopped until the next byte (the first byte in next instruction) is delivered from the memory.

Memory reference instructions

The operation of instructions having OP part 5 is partially determined by the argument. The third microinstruction at address 1B is therefore a jump determined by the argument so that the program can continue in 16 different ways. However, only four different operations shall be carried out and thus the next jump results in one of four addresses. At these addresses the "working" part of the microprogram (for load and increment, load and decrement, store and increment and store and decrement) start.

The argument is used to address registers via register address circuits. In these circuits, the necessary modifications are made, controlled by e-type microinstructions. For both SIN and SDE the most significant bit (Arg:3) must be zero-set at the output (reg address:3). As the memory addresses are stored in two register (a register pair), the least significant bit must be changed to transfer the complete address to the memory address register MAR (and also while incrementing or decrementing the register content).

After the load or store is performed, the contents of AR must be put into the stack, since AR must be used when incrementing or decrementing. The memory must be provided with a sufficiently long write pulse. This is obtained by inserting two microinstructions (the stack affecting microinstructions) between the setting and resetting of write MEM.

The increment-part (INC) of the microprogram in common to LIN and SIN and the "decrement-part" (DEC) is common to LDE and SDE. Both the INC and DEC starts with the least significant part of the address, which is stored in the odd register.

INC

If the number in the odd register becomes zero, also the most significant part of the address, which is stored in the even register must be incremented. The carry and zero flip-flops are affected while the a-type instruction at address 37 is executed, which means that the states of these flip-flops are determined by the result of the entire (16-bit) number.

DEC

For normal decrements (when the odd register content is not zero) the microinstructions at the following addresses will be executed: 40, 41, 42, 4B, 4C, 4D. If the odd register content becomes zero after the decrement, also the even register must be checked (4E, 4F). If this contains zero, the

zero FF will be one-set else it will be zero-set. (Carry flip-flop is one-set in all these cases).

If the odd register contains zero from the outset, both registers must be decremented (40, 41, 43, 44, 45, 46). In this case the content of the odd register becomes FF. If the content of the even register also becomes FF, zero flip-flop becomes zero-set and carry flip-flop becomes zero-set. For other results in the even register zero FF is zero-set and carry FF is one-set. Then the micro program continues at addresses 47 and 48. However, if the content of the even register becomes 00, the zero flip-flop becomes one-set, which is wrong. Then the micro program continues at addresses 47, 49 and 4A, where with the zero flip-flop becomes zero-set as it should be since AR is not affected at address 46. (Carry is one-set in the last case.)

Instructions having OP part 4

The instructions having OP part 4 perform operations of a very different character. Like for OP5, the third microinstruction (address 66) is a jump determined by the argument. Several instructions can then be executed by executing a single "working" microinstruction, but for some instructions another jump (Mij or Mir) must be made in the microprogram. Since the relative jumps (Mir) can be made to only each fourth address, some addresses are "empty" (e.g. 77 and 85).

Instruction 43, MORR ab

After the jump (Mij) in the fourth microinstruction has been performed, the MIC holds address B9. First the contents of AR is put into the stack, then the stack counter is decremented (BA). Then the next byte is entered into IR (BB).

The 4 LSB of the second byte are now used as register address (BC), which means that the contents of register b is transferred to AR. The microinstruction at address BD shifts IR one step right, wherewith the 4 MSB ("OP part" of the second byte) is set in "argument" position. Thus the microinstruction at address BE transfer AR (that holds the contents of register b) to register a.

Instruction 48, DRCM

The "working" part (starts at C0) of instruction 48 uses the same principle as described above (43). Here, the register contents are only compared, but as the comparison embraces two register pairs, the microprogram is quite long. First the content of the 8 LSB of register pair p1 is put into AR (C2–C5) and compared to the 8 LSB of p2 (C6–C7). Then preparations for comparing the 8 MSB is made (C8–CB; p1 in AR). If the first comparison shows equality (CC), the comparison of the 8 MSB determines the result (CE and CF).

If the first comparison (8 LSB) shows that the content of p1 > p2 (Carry = 1, Zero = 0; CC, CD, DO), the 8 MSB are compared (D2). This comparison will, of course, give the complete result, but if zero flip-flop is one-set it must be remembered that unequality is already discovered why zero flip-flop must be zero-set (D3, D4).

If the first comparison (8 LSB) shows that the content of p1 < p2, a jump is made (carry = 0, zero = 0; CC, CD, D0, D1, D6). In this case, both carry and zero flip-flop must be zero-set if the second comparison (D6) gives equality (due to the unequality found in the first comparison; program D7, D9, DA, DB). For all other results, the second comparison gives the complete result.

Instructions having first byte 42

The first byte causes the MIC to give following addresses 0, 1, 66, 74, 80, 8E to 8F, where the second byte is entered into the IR. Then the 4 LSB determines a relative jump (90). Then another jump to the "working" part of the microprogram is made.

The instruction 42 p4 (DRINp) and 42 p5 (DRDEp) have the same type of microprogram. First the register address is set in argument position, then register address LSB is set to one. After that, the incrementing or decrementing part of the microprograms belonging to instruction having OP part 5 is used.

The instructions 4248 (ROLE), 4258 (RORI), 42a9 (ADCYa) and 42aB (SBCYa) all will pass addresses EC–EE, where the IR is shifted one step left and the mark (carry) flip-flop is one-set. The IR shift will cause the following changes (see Mie 0 and Mie 8 in section entitled "Microinstruction type e"):

- 48 becomes 44
- 58 becomes 45
- a9 becomes Ca
- aB becomes Da

Then a jump is made to the start addresses for OP4, OPC or OPD. The AR shift will then be performed as in 44 and 45, but carry is shifted in. The addition and subtraction will be performed as in Ca and Da, but carry flip-flop is considered while calculating. The mark (carry) flip-flop makes the necessary modification.

Interrupts

The interrupt signals arriwing from the "peripheral" units sets a flip-flop in the interrupt register. The flip-flop outputs are arranged in priority order in to the priority encoder. This encoder gives the address to the one of the set flip-flops that has the highest priority. The encoder is disabled if instruction disable interrupt (4D) has been executed or after instruction change extension bit (IC:12) until instruction jump or call has been executed.

When the last microinstruction of an instruction is executed (i.e. when an instruction completely executed), the address is entered into the interrupt address register. Then the priority logic sends the signal interrupt to Mij, Mir, end decoder. The MIC then is incremented by one as selector 2 selects the adder outputs. Simultaneously, the two microinstructions at addresses 0 and 1 are inhibited.

The MIC is consequently stepped to address 2, while the instruction counter still holds the address to next instruction. Then the IC contents is put into the stack and the interrupt address is entered into the IC (3 and 4). The most significant part of the IC is zero-set (see Mie 3 in section entitled "Microinstructions type e"). At the same time, the decoder gives a reset signal to the interrupt flip-flop that has caused the interrupt.

The priority logic registers the priority level of the program that is started. If another interrupt signal arrives, the priority logic determines if the "started" program shall be interrupted. If so, the priority logic sees to it that the "started" program is restarted when the other higher priority program is finished.

One of the interrupts comes from the timer. If this interrupt is not reset within a certain time, the MPU circuits are resetted as at power on.

Timing

Most of the timing signals within the whole control unit are derived from a free-running oscillator in the MPU. The oscillator frequency 12,288 MHz is divided down to obtain signals with frequencies down to 0.6 Hz.

For use interally in the MPU three clock signals CP0–CP2 are generated in the CP counter. These clock signals controls that all events described in the microprogram occur at proper time (see Fig. 2). The CP counter can be stopped in several ways, thus stopping the MPU operation. The MPU generally operates continuously and the execution of an instruction starts immediately when the previous is finished.

Every time the memory address is changed, an access timing counter is triggered. If an attempt is made to use the memory output before the access time has elapsed, the CP counter is stopped. It can also be stopped by externally generated signals (e.g. generated in control panels). The stop position is after CP1, see Fig. 2. This means that the control signals will not be applied until when the CP counter is started again (this is valid for most of the control signals, as they are applied during CP0).

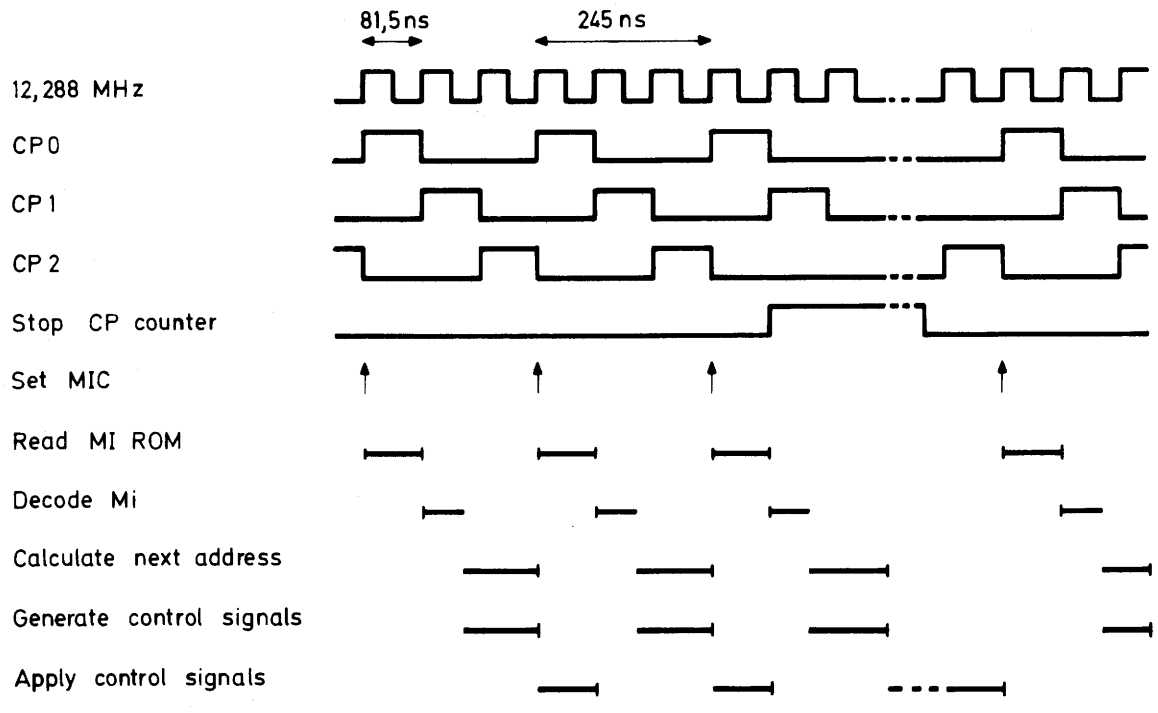


Fig. 2. MPU timing

The length of control signals generated in the MPU is determined by the number of microinstructions between setting and resetting of the signal (it is also possible to make slight adjustments by taking another of the three CP's). In some cases it is therefore necessary to insert no op (no operation microinstructions).

MPU relations to other units

The MPU controls the operation of the control unit except display logic (DIL). The interrupt signals and condition signals may come at arbitrary time. However, the display logic must have access to the memory when the characters for one line shall be transferred to the line store. The display logic program is determined by the presentation principles, and cannot be depending of the MPU operation.

To allow the display logic access to the memory, the MPU operation is stopped. It is made in different ways in single configuration and cluster configuration.

In single configuration, the access timing counter is kept preset by a signal (MEM occupation) from display logic. Then MPU can operate until it attempts to read the memory. The signal MEM occupation arrives in advance to the DIL access. Immediately before the DIL puts the output from the memory address counter (MAC) in the display adapter on the MEMA bus, it sends signal MEMA from display to the MPU. Then the MEMA selector is disabled.

In cluster configuration, the display memories and printer memories are mounted on the display adapter and the printer adapter respectively. The MPU operation can then proceed as long as the MPU does not need access to display or printer memory. If the MPU makes an attempt to read or write in these memories, the MPU operation is immediately stopped by the signal stop CP counter.

DETAILED DESCRIPTION

The operation of the circuitry is in many cases evident if the circuit diagrams are read in connection with the functional description given in the previous pages. In the following detailed description only some additional remarks are given.

Instruction register

The operation of the IR (MPU1 sh 4) is controlled by the two mode control signals IR MC0 and IR MC1 arriving from the Mie decoder on MPU2 (sh 4). The following list shows the operation

Mie	IR MC0	IR MC1	IR operation	
0	L	H	Shift left	Arg → OP
1	H	H	Parallel load	MEMO → IR
8	H	L	Shift right	OP → Arg
all other	L	L	No change	

Selectors

The operation of the selectors is evident from the logic symbol. If high address inputs are interpreted as ones and A0 as least significant bit, the number at the address inputs indicates which data input is selected. Example: (Condition selector, 74 on MPU1 sh 4). Arg = D gives A1 = 1, A0, 2 and 3 = 0 which means input B2 selected (compare instruction 2D XX – which calls for a relative jump if the printer is inoperable – and signal to B2).

Selector 1

Selector 1 (MPU1 sh 4) is blocked when a t-type microinstruction is at hand (Mi:7) wherewith the address to the CONROM is 00000, see Appendix 2. The Mij decoder gives the signal select OP (high) if Mi:0 through 3 are high. Thus the address to the CONROM is determined by the OP part, see Appendix 1 (page 3 Mia and page 4 Mij).

Microinstruction counter (MIC)

Since both mode control inputs are high, the MIC (MPU1 sh 4) is loaded each time CP0 goes high. The address can arrive to the input from different sources, see the following table. The MIC is zero-set by CP and when the last microinstruction in an instruction is executed.

Microinstruction type	Mij CONROM (3)	Mij start (1)	Mij start dis (2)	Selector 2 (21, 9, 17, 5)	Source	Remark
Mit, Mie, Mia	H	H	L	B3–C3	MIC+1	Selector 3 selects D0–E0 (all zero, Mir gates closed), skip selector blocked – output high, thus adders add one to old MIC value
Mij 1111 (no int)	H	L	L	B1–C1	CONROM 18	Address to CONROM determined by OP part. Start addresses in 18
Mij 1111 (int)	H	L	H	B3–C3★	MIC+1	Selector 3 and skip selector blocked. 66 adds 1. Used to reach interrupt microprogram
Mij 1110	H	H	L	B3–C3	MIC+1+$\overline{\text{Arg}}$	Selector 3 selects B1–E1 ($\overline{\text{Arg}}$). Skip selector output high
Mij 0000–1101	L	H	L	B2–C2	CONROM 19	Address to CONROM determined by microinstruction (Mi:0–3). Label addresses in 19
Mis	H	H	L	B3–C3	MIC+1+CS	Selector 3 selects B0–E0 (E0 one) thus 66/Cin = 1. Skip selector output according to condition under test
Mir	H	H	L	B3–C3	MIC+1+4 · · $\overline{\text{CBA}}$	Selector 3 selects B0–E0. Mir gates open. LSB enters adder B2 (thus 4x). Skip selector output high

★ half the selector (21,9) is blocked

Type decodes and latches

When the clock signal (CP0, low polarity) at the inputs (39/1, 15/1, 40/1, see MPU1 sh 4) is high, data from MI ROM is loaded into the priority decoder 74278 that serves as type decoder. One of its outputs is high, indicating the most significant one bit in the microinstruction at hand. All outputs are low if the overriding Pin is high (indicating Mit). If no one is present in outputs 3–7 of MI ROM (28/9 – 27/9), Pout is low indicating Mir.

The selectors 15 and 40 feeds the output from MI ROM to their outputs while CP0 (low polarity) is high and latches the output when CP0 goes low as the outputs are feed to B0–E0 inputs.

Timing circuits

The frequency of the free-running oscillator is 12,288 MHz, which gives a period time of 81,5 ns. This frequency is then divided as indicated in the circuit diagram MPU1 sh 3 to obtain suitable frequencies for several other circuits.

The operation of the CP counter is illustrated in Fig. 2 (functional description page 13).

The access timing counter is loaded each time the memory address (MEMA) is changed. The value can be chosen to match different memory access times. The counter is then incremented until carry occurs. Then gate 69/9 is closed and the signal read MEMO cannot stop the CP counter any longer.

Interrupt circuits

Interrupt register

The interrupt register (MPU1 sh 3) contains seven flip-flops. These work independently of each other, except that all other flip-flops are reset when the power-on flip-flop is set. The states of six of the flip-flops are strobed into the register 4 by the signal CP end (that arrives when an instruction is executed). This is made because the interrupt signal can arrive at any time and cause a change in the priority encoder when its outputs are read.

Priority encoder

The priority encoder (14 on MPU1 sh 3) outputs give a binary number (low = 1) determined by the low input that has highest priority, provided that Pin (14/5) is low. The GS output (14/14) is low if any input is low, thus indicating that an interrupt is requested. If Pin is high, all outputs are high. Pin can thus be used to disable interrupts. The binary number from the encoder is used as interrupt address and is stored in 26 along with the signal interrupt requested. Example if power on FF is set, 49/6 is low wherewith outputs 6, 7, 9 and 14 of encoder 14 are low. Thus P2/6, 8 and 34 are high, which is interpreted as address 000X.

Priority logic

The priority logic, which determines whether the requested interrupt shall be executed or not, incorporates two flip-flops (50) and several gates (MPU1 sh 3). Output 6 (test point 40) is high when a program is running on priority level 0, and output 8 (test point 39) is high when a program is running on priority level 1. If a program is to be interrupted, inverter 73/8 sends out the signal interrupt to the Mij decoder.

The signal interrupt is sent in the following three situations:

- The program is running on the background level (i.e. both test points 39 and 40 are low), when an interrupt signal "to level 1" is registered (i.e. test point 38 is low). In this case gate 61/4 is high.

- The program is running on the background level when an interrupt signal "to level 0" is registered. In this case also gate 61/1 is high
- The program is running on level 1 (test point 39 is high) when an interrupt signal "to level 0" is registered. In this case gate 61/1 is high.

Then the microprogram proceeds to the interrupt routine. One of the microinstructions causes a pulse to be sent to inputs 1 and 9 to gates 38. In the first of the three situations 50/8 goes high, in the other two 50/6 goes high.

The last instruction of the program on the high level contains a microinstruction that issues a clock pulse to the two flip-flops in 50. Then the flip-flops are reset to original position except when both 50/6 and 50/8 are high (as they will be in the third situation). In this case 38/6 is low, keeping 50/8 high.

Timer

The timer incorporate flip-flops 48 and shift register 47 (MPU1 sh 3). Each clock interrupt signal shifts a one (via 48/9) through 47. When the clock interrupt is executed 48/9 and 47 are zero-set. If the clock interrupt is not executed before nine positive transitions in the clock interrupt signal, 48/6 goes low whereupon flip-flops 71 and gate 57/8 issues a pulse (approx 13 μs long).

Reset circuit

If the +5 V supply fails, or if the reset button is depressed, gate 58/8 goes low (MPU1 sh 3). It is also low some time after power on due to the capacitor C1. Then gate 57/8 is low. This output is used as general reset (GR intern) in MPU1. Among other things GR intern sets the power-on / / reset flip-flop in the interrupt register. Via 57/3, GR is sent to other PC boards as long as this flip-flop is set.

The microinstruction counter is running through addresses 0–8 as long as GR intern is low (as 50/5 and 50/9 are kept high). The power-on / reset flip-flop is cleared by the microinstruction at address 5. This means that the trailing edge of the GR signal sent to other units comes at a well defined time.

Condition flip-flop circuitry

The zero and carry flip-flops are mounted on MPU1 but some associated selector circuits are mounted on MPU2. Fig. 3 shows an excerpt of two circuit diagrams. The selector 77 on MPU1 also selects mode control signals for the AR. In Fig. 3 its symbol is slightly rearranged according to the operation.

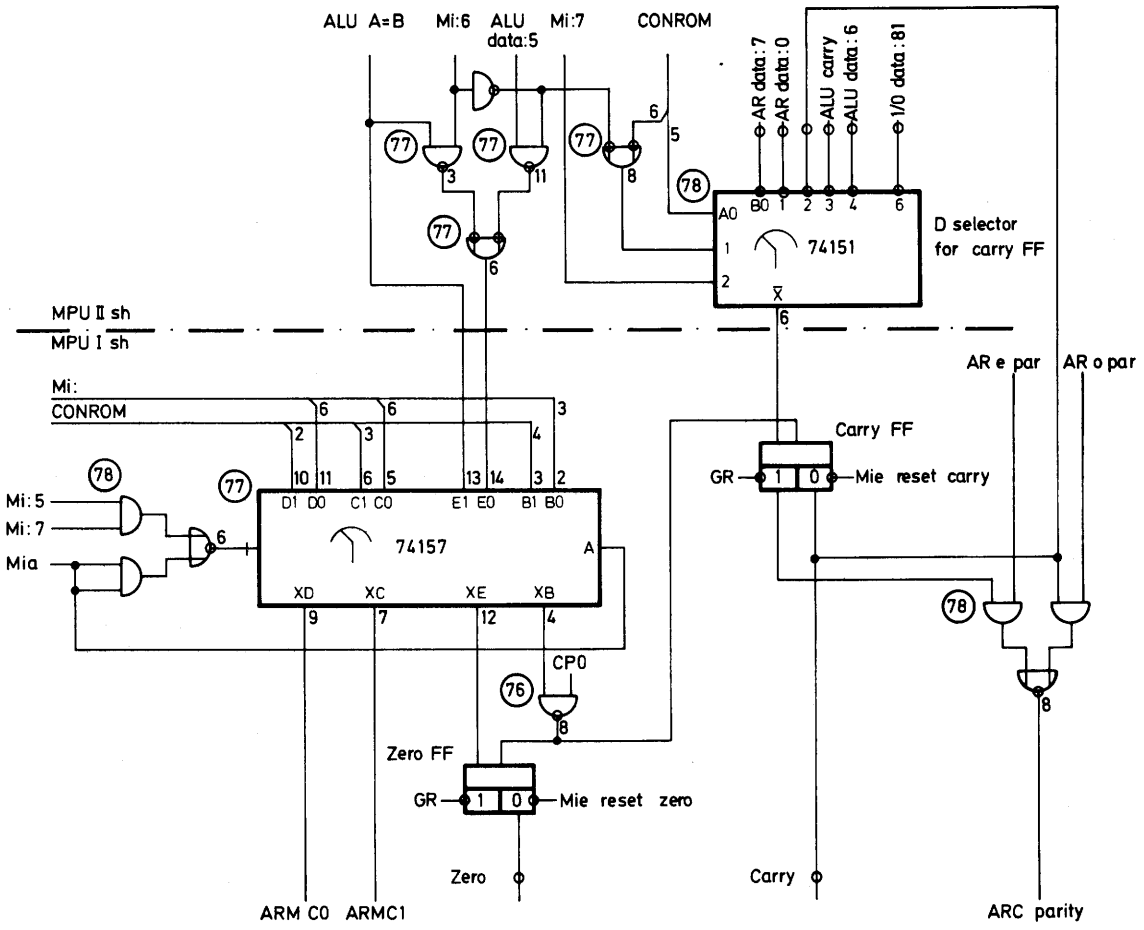


Fig. 3. Carry FF and zero FF

AR mode control selector

When an a-type microinstruction is executed, CONROM:2 and 3 are used as AR MC0 and AR MC1 respectively. For two destinations when a t-type microinstruction is executed (Mitd 6 and 7), the AR shall be loaded. In these two cases Mi:5, 6 and 7 are all ones i.e. AR MC0 and AR MC1 are both high. For all other microinstructions selector 77 is blocked i.e. its outputs are low.

Clock pulses to zero and carry FF

The XB output of selector 77 is high when an a-type microinstruction is executed provided CONROM:4 is high (see Appendix 2). It is also high in connection with Mitd 3 and Mitd 7 (Mi:3, 5 and 7 are high). Only in these cases clock pulses are permitted to pass gate 76/8.

D selector för zero FF

When an a-type microinstruction is executed, zero FF is set according to ALU A = B. When a t-type microinstruction including Mitd 3 is executed zero FF is set according to ALU data:5 (as Mi:6

is low, Mi:5 and Mi:7 are high). When a t-type microinstruction including Mitd 7 is executed, zero FF is set according to ALU A = B (as Mi:5–7 are high).

D selector for carry FF

The operation of the selector feeding D-input of carry FF is summarized in the following table

Mi: 6 7	CONROM 5 6	Selected input	Remarks
0 0	0 0	Carry	Mia 1, 8 (carry FF shall not be affected)
0 0	0 1	AR data:7	Mia 2 (AR shift left)
0 0	1 0	ALU carry	Mia 6, 7, 9–14
0 0	1 1	AR data:0	Mia 3 (AR shift right)
0 1	0 1	ALU data:6	Mitd 3
1 1	0 1	I/O data:81	Mitd 7

Note. The CONROM gives 01 for all t-type microinstruction, but selector position is only interesting in the two cases when clock pulses are permitted to carry FF. Similary, selector position is not interesting in Mia 0, 4 and 5 or in any Mie.

Parity check

The parity checker (on MPU2) can only check the number of ones in eight data bits. The carry FF and gates 78/8 are used to form the "overall" parity check condition for the condition selector. The operation is shown in the following table

No of ones in AR data	AR o parity	AR e parity	PC (I/O data:81)	Carry FF	ARC parity
even	L	H	0	0	H
odd	H	L	0	0	L
even	L	H	1	1	L
odd	H	L	1	1	H

Thus a low signal ARC parity indicates that the nine input bits (8 bit data + PC) have an odd number of ones.

ALU

The ALU (arithmetic and logic unit – MPU 2 sh 3) is composed of two DIP's 74181. Each of the DIP's can perform several different operations on two 4-bit words. The operation is selected by the four function select (FS – in appendix 2 denoted s) inputs and the MC input. The DIP's also have a carry input (Cin) but this is considered only when MC is low. The operations that are used in the

MPU are summarized in the following table (see also the table in Appendix 2 and Mia in Appendix 1).

Inputs FS3	FS2	FS1	FS0	MC	Cin	ALU operation	MPU operation
L	L	L	L	L	L	X = A−1	AR−1 → R
L	H	H	L	L	H	X = A−B	AR−R → AR, compare
H	L	L	H	L	L	X = A+B	AR+R → AR, IC+MEM → IC
H	L	L	H	H	irr	X = A⊕B	AR ex or R → AR
H	L	H	L	H	irr	X = B	R → AR, Mit
H	L	H	H	H	irr	X = A or B	AR or R → R
H	H	H	L	H	irr	X = A and B	AR and R → R
H	H	H	H	L	H	X = A+1	AR+1 → R
H	H	H	H	H	irr	X = A	AR → stack, AR →R, set Z

Note that the ALU operates in negative logic (i.e. one = low). Two of the other outputs are also used: Cout, which goes high if the arithmetic operation gives carry, and output A = B, which is wired-or gated and goes high if all outputs become zeroes (i.e. high).

AR

The operation of the AR (accumulator register – MPU 2, sh 3) is evident from the table below (see also section entitled AR mode control selector, page 19).

CONROM:2 AR MC0	CONROM:3 AR MC1	AR operation	Remark
L	H	Shift left	Mia 2 (instruction 44)
H	L	Shift right	Mia 3 (instruction 45)
H	H	Parallel load	Mia (→ AR), Mitd 6 and 7
L	L	No change	

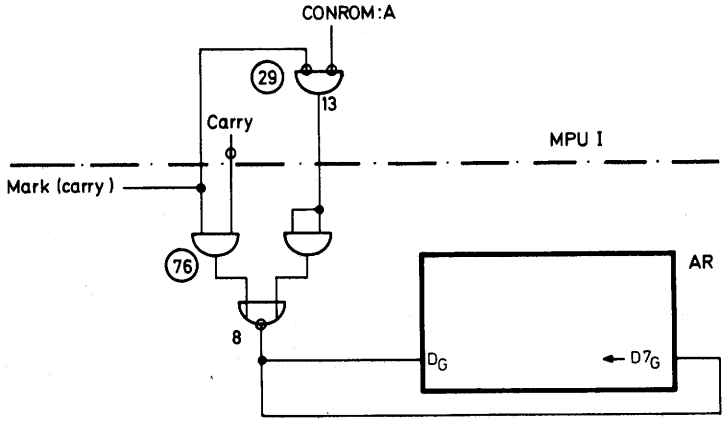


Fig. 4. Shifting conditions

When shifting, zero is entered into the most (right shift) or least (left shift) significant bit if mark (carry) is low (FF zero-set). CONROM:A is low for Mia 2 and Mia 3. If mark (carry) FF is one-set, the contents of carry FF is shifted in instead.

Output 76/8 is also feed to ALU Cin. Thus the carry condition for the ALU can also be selected by the mark (carry) FF.

Stack and register

RWM DIP's

In both the register and the stack, SN7489 circuits are used (MPU 2 sh 3). The SN7489 contains 64 flip-flop memory cells organized in a matrix to provide 16 words of four bits each. Thus two DIP's will provide a 16 byte memory. Each of the 16 words is addressed straight binary using the address inputs.

The operation on the addressed memory cells is determined by the MC and WE inputs according to the table

MC	WE	Operation	Condition of outputs
L	L	Write	Complement of data inputs
L	H	Read	Complement of addressed word
H	L	Inhibit storage	Complement of data inputs
H	H	Do nothing	High

Stack

The capacity of the stack is 48 bytes, provided by six RWM DIP's. The addressing circuits contains two counter DIP's, two decoders and several gates.

The counters are set to FF by general reset signal ($\dot{G}R$). That is done because GR returns to the high level when a microinstruction push stack (Mia 5) remains in the microprogram that is run at power on or reset. After that microinstruction the stack counters are zero-set.

Gates 64/8 and 30/3 control the WE input to the RWM DIP's. Gate 64/8 is controlled by Mia and CONROM:0 (push stack, see Appendices 1 and 2) and the other input to 30/3 goes low when a microinstruction involving Mitd 0 is executed. The stack counter is always incremented after a write operation. Gate 29/11 delays the write enable pulse so that it will not arrive too much before the MC pulses (through 30/8, 58).

Decoder 45 output 9 (X7) goes low when Mi:7, 0 and 1 are one and Mi:2 is zero, which in the case when a t-type microinstruction (Mits 3) is executed. When gate 30/8 goes low, decoder 58 enables two of the RWM DIP's via their MC inputs.

Register

The register comprises four RWM DIP's, giving a capacity of 32 bytes, divided into two blocks of 16 bytes each. Each of the 32 bytes can be selected by the argument (via register address circuits) and the register block FF.

The gate 64/12, that sends the WE pulse to the register DIP's, is controlled by CP0, Mia and CONROM:1 (see Appendices 1 and 2).

The MC input is controlled by the register block FF, several gates and decoder 45. Gate 32/6 goes low for all Mia except Mia 4 (as CONROM:7 is high for Mia 4 – and 65/3 is then low) provided that "MEM to B Mia" FF is not set (if it is set, input 3 to 32 is low). Decoder 45 output 7 (X6) goes low when Mi:7 and 1 are ones and Mi:2 and 0 are zeroes (Mits 2).

The register address circuits comprises three flip-flops and three gates. These circuits can affect the MSB (reg address:3) and the LSB (reg address:0). Microinstruction Mie 24 (M0) sets FF9–21 so that gate 39/6 is closed giving a zero in reg address:3. (Arg and reg address are in negative logic.) Microinstruction Mie 25 (L0) closes gate 27/6 while Mie 26 (L1) open it.

B-selector

Data from several sources can be put onto the B-bus. When any t-type microinstruction is executed, data passes via the B-bus and the ALU. Most of the a-type microinstructions also call for a data transfer via the B-bus. Data from other sources pass the B-selector that consists of selectors 84 and 72 (top of MPU 2 sh 3) and selectors 51, 74, 63 and 40 (bottom of MPU2 sh 3).

Selector address signals are generated in the Mits decoder, that comprises decoder 45 and several gates. Some of these gates are also affected by Mia. The Mits decoder also controls the stack and the register, see above. The relevant states of the Mits decoder are shown in the following table

Mi	Decoder 45	Gates 34		
0 1 2 7	5 6 7 9	6 3 11	Source	Mits
0 0 0 1	L	H	I/O data	0
1 0 0 1	L	H	MEMO	1
0 1 0 1	L	H	Reg	2
1 1 0 1	L	H	Stack	3
0 0 1 1		L L L	ICL	4
1 0 1 1		L H L	RCZ ICM	5
0 1 1 1		H L L	New IC:12, Reg address (= Arg)	6
1 1 1 1		H H L	Interrupt address	7

From the table it is evident that decoder 45 is used for Mits 0–3 and gates 34 for Mits 4–7. In the former case selectors 84 and 72 are opened (only Mits 0 and 1), in the latter selectors 51, 74, 63 and 40. Selectors 84 and 72 are also opened (in pos MEMO) if "MEM to B Mia" flip-flop is set. Selectors 51, 74, 63 and 40 are also opened (in pos ICL) if a Mia with CONROM:7 high is executed.

Gate 46/8 sends out the signal read MEMO, used to check the memory access timing, in the following situations

- Mie 1 set IR, increment IC (MEMO is loaded into IR)
- "MEM to B Mia" FF is set (MEMO is to be read in next microinstruction)
- Mits 1 (MEMO is source for data to B-bus)
- Mia and CONROM:7 high (MEMO is added to ICL)

Mie decoder

The Mie decoder comprises three decoder DIP's and one selector (MPU 2 sh 4). The decoder 8 operates as a group decoder:

- output 4 (XB3) is low for Mie 16–31
- output 5 (XB2) is low for Mie 0–15
- output 11 (XC2) is low for Mie 0, 1, 2 and 9

Decoders 4 and 6 are enabled during CP0 for Mie 16–31 and Mie 0–15 respectively. They provide decoding of all 32 Mie using Mi:0–3 as inputs.

Selector 41, which is not depending on any timing pulse, gives mode control signals to IR (outputs 4 and 5) and a signal sent to gate 46/8 (see last paragraph in previous section). The operation of the selector is tabulated below.

Mie	Mi:0	Mi:$\overline{0}$	Mi:3	41/4	41/7 IRMC1	41/9 IRMC0
0	0	H	0	L	H	L
1	1	L	0	H	H	H
8	0	H	1	L	L	H
9	1	L	1	L	L	L

Instruction counter (IC)

The IC comprises three counter DIP's that gives 12 bits. The 13th bit is stored in flip-flop 16/3 (MPU2 sh 4). Note that the address contained in IC is in negative logic, which means that the input T – must be used to increment the counter content. Note also that ICL is loaded with ALU data via MARL.

The most significant bit IC:12 is not changed directly. Instead an auxiliary flip-flop (change IC:12) is set. At the next microinstruction involving Mitd 2, the new value of IC12 (calculated in gates 28/6) is loaded into 16/3 via B-bus and ALU. The new value of IC:12 is calculated as shown in following table

IC:12	16/5	16/9 (change FF)	28/6	New IC:12	Remark
0	H	H	L	1	When 16/9 is high, change
0	H	L	H	0	is requested
1	L	H	H	0	
1	L	L	L	1	

The most significant bit can be one-set during incrementation if both ICM and ICL gives carry.

The circuit comprising flip-flops in 62, gates in 50 and 39 makes it possible to perform relative jumps both forwards and backwards. In a relative jump the eight bits from the A-selector (they can come from MEMO or AR data) are added to the eight bits from ICL. The most significant bit on the A-bus (A-bus:7) indicates whether the number is positive (A-bus:7 is zero) or negative (in two's complement). The operation is an follows

- If A-but:7 is zero (= positive number), 62/7 goes high when CP2 goes high (see Fig. 2 on page 13). Then

  a) if 62/10 went high because the addition in ALU yielded carry, 50/8 is low. ICM will then be incremented at CP0.

  b) if 62/10 went low, both 50/8 and 50/6 are closed. ICM will not be affected.

- If A-bus:7 is one (= negative number), 62/6 goes high when CP2 goes high. Then

  c) if 62/10 went high, both 50/8 and 50/6 are closed. ICM will not be affected.

  d) if 62/10 went low, 50/6 is low. ICM will then be decremented at CP0.

MICROINSTRUCTIONS

Format

There are six types of microinstructions. They always contain eight bits. The most significant bits are used for classification, see Fig.

Type	Mibit 7 6 5 4 3 2 1 0		
Mit	1 end	transfer information	If Mi:4 "end" = 1 next microinstruction is fetched from memory cell with address 0.
Mie	0 1 end	execute	
Mia	0 0 1 end	ALU operations	
Mij	0 0 0 1	jump instructions	
Mis	0 0 0 0 1	skip instructions	
Mir	0 0 0 0 0	relative jump	

Microinstructions type transfer (Mit)

Format "1" in Mi:7 7 6 5 4 3 2 1 0

1 F E end D C B A

destination source

F E D	Mit d number	Destination	
0 0 0	0	Stack	(stack counter is incremented after entry)
0 0 1	1	ICL	(8 least significant bits)
0 1 0	2	ICM	(5 most signf. bits); change IC12 FF is reset
0 1 1	3	RCZ ICM	(Carry from ALU data:6, zero from ALU data:5)
1 0 0	4	MARL	(Memory address register, least significant part)
1 0 1	5	MARM	(most signf. part). (MEMA is taken from MAR until MEMO is read once or until a Mie (1 or 12) is executed
1 1 0	6	AR	
1 1 1	7	AR	Carry FF and Zero FF are also set (C from I/O, Z from ALU:A = B)

C B A	Mit s number	Source		
0 0 0	0	I/O data	(according to arg)	
0 0 1	1	Memory		
0 1 0	2	Register	(according to Reg bl FF and arg)	
0 1 1	3	Stack	(Note! stack counter is not affected)	
1 0 0	4	ICL		
1 0 1	5	RCZ ICM		
1 1 0	6	0 0 0 N	Arg from IR	(N = New IC:12)
1 1 1	7	0 0 0 0	X X X 0	X X X = Interrupt address

Microinstructions type execute (Mie)

Format "0" in Mi:7, "1" in Mi:6 7 6 5 4 3 2 1 0

0 1 E end D C B A

E	D C B A	Number	Operation	
0	0 0 0 0	0	Shift IR: change arg → OP	
0	0 0 0 1	1	Set IR. Increment instruction counter (if no interrupt).	
0	0 0 1 0	2	Increment instruction counter	
0	0 0 1 1	3	Set ICM to F (15), set pri, reset interrupt FF	
0	0 1 0 0	4	Set change IC:12 (IC:12 is changed at next Mit d2, whereupon change IC:12 is reset. Interrupts are inhibited until then.)	
0	0 1 0 1	5	Set reg block FF to select block 0	
0	0 1 1 0	6	Set reg block FF to select block 1	
0	0 1 1 1	7	Decrement stack counter by 1	
0	1 0 0 0	8	Shift IR: change OP → arg	
0	1 0 0 1	9	Reset (initiates GR signal that 0-sets: change IC:12, C, Z, stack counter, interrupt FFs and causes interrupt).	
0	1 0 1 0	10	Spare	
0	1 0 1 1	11	Spare	
0	1 1 0 0	12	MEMA selector is set to IC	
0	1 1 0 1	13	Increment ICM	
0	1 1 1 0	14	Set B to MEM for following Mia	(reset only by end)
0	1 1 1 1	15	Set Write MEM	
1	0 0 0 0	16	Disable interrupt	
1	0 0 0 1	17	Enable interrupt	(enabled also by GR)
1	0 0 1 0	18	Reset priority logic	
1	0 0 1 1	19	Reset Write MEM	(reset also by GR)
1	0 1 0 0	20	Set I/O data to MPU	(reset only by end)
1	0 1 0 1	21	Set enable I/O data	(reset only by end)
1	0 1 1 0	22	Set I/O data from MPU	
1	0 1 1 1	23	Reset I/O data from MPU	(reset also by end)
1	1 0 0 0	24	Set Reg address: 3 to 0	(reset only by end)
1	1 0 0 1	25	Set Reg address: 0 to 0	} (cancelled by end)
1	1 0 1 0	26	Set Reg address: 0 to 1	
1	1 0 1 1	27	Reset Zero FF	

E	D	C	B	A	Number	Operation	
1	1	1	0	0	28	Reset Carry FF	
1	1	1	0	1	29	Set Mark (carry)	(reset also by Mie1)
1	1	1	1	0	30	Reset Mark (carry)	
1	1	1	1	1	31	Spare	

Microinstruction type ALU operation (Mia)

Format "0" in Mi:7 and Mi:6, "1" in Mi:5 — 7 6 5 4 3 2 1 0

0 0 1 end D C B A

Control signals to ALU are generated in CONROM (address to CONROM = 0 $\overline{D}$ $\overline{C}$ $\overline{B}$ $\overline{A}$; or = $\overline{OP}$ if A = B = C = D = 1)

D	C	B	A	Address to CONROM	Number	Operation
0	0	0	0	0 1 1 1 1	0	AR → Reg (Reg address from arg and R)
0	0	0	1	0 1 1 1 0	1	Set Zero to 1 if AR = 0, else to 0
0	0	1	0	0 1 1 0 1	2	Shift AR left AR data:7 (MSB) → Carry
0	0	1	1	0 1 1 0 0	3	Shift AR right AR data:0 (LSB) → Carry
0	1	0	0	0 1 0 1 1	4	ICL + MEMO ☆ → ICL, increment ICM if ALU carry
0	1	0	1	0 1 0 1 0	5	Push stack ie AR → stack, increment stack counter
0	1	1	0	0 1 0 0 1	6	AR+1 → Reg (Reg address from R and arg). Carry and zero are set according to ALU:Cout and ALU:A = B resp.
0	1	1	1	0 1 0 0 0	7	AR−1 → Reg (as above). Carry, zero as in Mia 6
1	0	0	0	0 0 1 1 1	8	Reg (as above)☆☆ → AR (load). Carry is not affected, zero as in Mia 6.
1	0	0	1	0 0 1 1 0	9	AR and Reg (as above)☆☆ → AR. Carry and zero as in Mia 6
1	0	1	0	0 0 1 0 1	10	AR or Reg (as above)☆☆→ AR. Carry and zero as in Mia 6
1	0	1	1	0 0 1 0 0	11	AR exor Reg (as above)☆☆→ AR. Carry and zero as in Mia 6
1	1	0	0	0 0 0 1 1	12	AR+Reg (as above)☆☆ → AR. Carry and zero as in Mia 6
1	1	0	1	0 0 0 1 0	13	AR−Reg (as above)☆☆ → AR. Carry and zero as in Mia 6
1	1	1	0	0 0 0 0 1	14	AR−Reg (as above)☆☆. Note AR is not affected. Carry and zero as in Mia 6. Used as compare.
1	1	1	1	0 $\overline{OP}$		Operation as above for different values of OP. Note that Mia 4 "ICL + MEMO → ICL" is changed to ICL + AR → ICL (if OP = 4 ie 0 1 0 0)

☆ Mia 4 is the only microinstruction when A-selector is in position MEMO. It is for all other microinstructions in position AR data.

☆☆ If flip-flop MEM to B Mia is set, information is taken from MEMO instead of Register.

MPU

Microinstructions type jump (Mij)

Format "0" in Mi:7, Mi:6, Mi:5, "1" in Mi:4 — 7 6 5 4 3 2 1 0

0 0 0 1 D C B A

D C B A	Designation	Operation
1 1 1 1	Mij start	Jump to start address for current OP part. The start address is provided from CONROM from cell with address 1 $\overline{OP}$ (5 bits)
	Mij start dis	If any interrupt FF is set, the jump to start address is inhibited
1 1 1 0		Microinstruction counter is incremented by $\overline{arg}$+1
0 0 0 0 through 1 1 0 1	Mij CONROM	Jump to address obtained from CONROM at address 1 $\overline{D}$ $\overline{C}$ $\overline{B}$ $\overline{A}$

Microinstructions type skip (Mis)

Format "0" in Mi:7, Mi:6, Mi:5, Mi:4 "1" in Mi:3 — 7 6 5 4 3 2 1 0

0 0 0 0 1 C B A

Next microinstruction is skipped if condition is fulfilled

C B A	Condition used for test
0 0 0	Zero FF
0 0 1	Arg = F
0 1 0	(CT) Condition from selector (governed by arg) is fulfilled (output high)
0 1 1	(CF) Condition from selector (governed by arg) is not fulfilled (output low)
1 0 0	Carry FF
1 0 1	OP:3
1 1 0	Mark (carry)

Microinstruction type relative jump (Mir)

Format "0" in Mi:7, Mi:6, Mi:5, Mi:4, Mi:3 — 7 6 5 4 3 2 1 0

0 0 0 0 0 C B A

Microinstruction counter is incremented by 4 x $\overline{C}$ $\overline{B}$ $\overline{A}$ + 1

1. Control Signals

Address	Operation	S3	S2	S1	S0	MC	Cin				A sel.	C sel.		CZ cp	mode sel.		AR	Wr reg	Push	
4 3 2 1 0		F	E	D	C	B	A	9	8	Hex	7	6	5	4	3	2		1	0	Hex
0 0 0 0 0	Mit	H	L	H	L	H				A8		H	L					L	L	40
0 0 0 0 1	Compare	L	H	H	L	L	H			64	L	L	H	H	L	L		L	L	30
0 0 0 1 0	AR − R → AR	L	H	H	L	L	H			64	L	L	H	H	H	H		L	L	3C
0 0 0 1 1	AR + R → AR	H	L	L	H	L	L			90	L	L	H	H	H	H		L	L	3C
0 0 1 0 0	AR ⊕ R → AR	H	L	L	H	H				98	L	L	H	H	H	H		L	L	3C
0 0 1 0 1	AR or R → AR	H	L	H	H	H				B8	L	L	H	H	H	H		L	L	3C
0 0 1 1 0	AR and R → AR	H	H	H	L	H				E8	L	L	H	H	H	H		L	L	3C
0 0 1 1 1	R → AR	H	L	H	L	H				A8	L	L	L	H	H	H		L	L	1C
0 1 0 0 0	AR − 1 → R	L	L	L	L	L	L			00	L	L	H	H	L	L		H	L	32
0 1 0 0 1	AR + 1 → R	H	H	H	H	L	H			F4	L	L	H	H	L	L		H	L	32
0 1 0 1 0	Push stack	H	H	H	H	H				F8	L			L	L	L		L	H	01
0 1 0 1 1	IC + MEM → IC	H	L	L	H	L	L			90	H			L	L	L		L	L	80
0 1 1 0 0	AR shift right						L				L	H	H	H	L	H		L	L	74
0 1 1 0 1	AR shift left						L				L	H	L	H	H	L		L	L	58
0 1 1 1 0	Set Z according to AR	H	H	H	H	H				F8	L	L	L	H	L	L		L	L	10
0 1 1 1 1	AR → R	H	H	H	H	H				F8	L			L	L	L		H	L	02
14 13 12 11 10	(DIP 18 and 19)	9	7	6	5	4	3	2	1 (DIP 18)		9	7	6	5	4	3		2	1 (DIP 19)	

Note: Unprogrammed outputs (empty positions in the tables) can be low (Intersil) or high (MMI and other manufacturers). The hexadecimal values given assume that empty positions are low.

2. Jump addresses

		Start addresses									Label addresses								
4	3	2	1	0	OP part	F	E	D	C	B	A	9	8	Hex	Label	7	6	5	4	3	2	1	0	Hex
1	0	0	0	0	OPF	L	L	L	H	H	L	L	L	18	–									
1	0	0	0	1	OPE	L	L	L	H	L	L	H	H	13	–									
1	0	0	1	0	OPD	L	L	L	H	L	L	H	H	13	–									
1	0	0	1	1	OPC	L	L	L	H	L	L	H	H	13	JCF	L	L	L	H	L	L	L	L	10
1	0	1	0	0	OPB	L	L	L	H	L	L	H	H	13	ASC	H	H	H	L	H	H	L	L	EC
1	0	1	0	1	OPA	L	L	L	H	L	L	H	H	13	Z = 0	H	H	L	H	L	L	L	L	D0
1	0	1	1	0	OP9	L	L	L	H	L	L	H	H	13	–									
1	0	1	1	1	OP8	L	L	L	H	L	L	H	H	13	DRCM	H	H	L	L	L	L	L	L	C0
1	1	0	0	0	OP7	H	H	L	H	H	H	L	L	DC	TR	H	L	H	H	H	L	L	H	B9
1	1	0	0	1	OP6	H	H	H	L	L	L	H	H	E3	LD	H	L	H	H	L	L	L	L	B0
1	1	0	1	0	OP5	L	L	L	H	H	L	H	H	1B	SDE	L	H	L	H	H	L	H	H	5B
1	1	0	1	1	OP4	L	H	H	L	L	H	H	L	66	SIN	L	H	L	H	L	L	L	H	51
1	1	1	0	0	OP3	H	H	H	L	H	H	H	H	EF	DEC	L	H	L	L	L	L	L	L	40
1	1	1	0	1	OP2	L	L	L	L	H	H	L	H	0D	LDE	L	L	H	H	H	L	L	H	39
1	1	1	1	0	OP1	L	L	L	L	H	L	L	H	09	INC	L	L	H	H	L	L	L	H	31
1	1	1	1	1	OP0	L	L	L	L	H	L	H	H	0B	LIN	L	L	H	L	H	L	H	H	2B

14	13	12	11	10	(DIP 18 and 19)	9	7	6	5	4	3	2	1	(DIP 18)		9	7	6	5	4	3	2	1	(DIP 19)

MICROPROGRAM

Address	Mi: 7 6 5 4	3 2 1 0	Mi hex	Mi type	Label	Operation
0	0 1 0 0	0 0 0 1	41	Mie 1		Set IR, increment IC if not interrupt
1	0 0 0 1	1 1 1 1	1F	Mij		Jump according to OP & CONROM if no interrupt
2	1 0 0 0	0 1 0 0	84	Mit		ICL → stack
3	1 0 0 0	0 1 0 1	85	Mit		RCZ ICM → stack
4	1 0 0 0	1 1 1 1	8F	Mit		Interrupt address → ICL
5	0 1 0 0	0 0 1 1	43	Mie 3		Load F to ICM, set pri, clear interrupt FF
6	0 1 0 0	1 1 0 1	4D	Mie 13		Increment ICM (to 0)
7	0 0 1 0	0 1 0 1	25	Mia		Push stack
8	0 1 0 1	0 1 0 1	55	Mie 5		Select reg block 0, end
9	1 0 0 0	0 1 0 0	84	Mit	OP1	ICL → stack
A	1 0 0 0	0 1 0 1	85	Mit		RCZ ICM → stack
B	1 0 0 0	1 0 0 1	89	Mit	OP0	MEM → ICL
C	1 0 1 1	0 1 1 0	B6	Mit		Arg → ICM, end
D	0 0 0 0	1 0 1 0	0A	Mis	OP2	Skip if condition true
E	0 1 0 1	0 0 1 0	52	Mie 2		Increment IC, end
F	0 0 1 1	0 1 0 0	34	Mia		IC + MEM → IC, end
10	0 0 0 0	1 0 1 1	0B	Mis	JCF	Skip if condition false
11	0 1 0 1	0 0 1 0	52	Mie 2		Increment IC, end
12	0 0 1 1	0 1 0 0	34	Mia		IC + MEM → IC, end
13	0 0 0 0	1 0 0 1	09	Mis	OP8-E	Skip if arg = F
14	0 0 1 1	1 1 1 1	3F	Mia		According to OP & CONROM, end
15	0 1 0 0	1 1 1 0	4E	Mie 14		Set MEM to B FF
16	0 0 1 0	1 1 1 1	2F	Mia		According to OP & CONROM
17	0 1 0 1	0 0 1 0	52	Mie 2		Increment IC, end
18	0 0 0 0	1 0 0 1	09	Mis	OPF	Skip if arg = F
19	0 0 1 1	0 0 0 0	30	Mia		AR → Reg, end
1A	0 1 1 1	1 1 1 1	7F	Mie 31		No op, end
1B	0 0 0 1	1 1 1 0	1E	Mij	OP5	Jump by adding $\overline{arg}$+1
1C	0 0 0 1	0 1 0 1	15	Mij	(5F)	Jump according to CONROM (SDE)
1D	0 0 0 1	0 1 0 0	14	Mij	(5E)	– " – SIN
1E	0 0 0 1	0 1 0 1	15	Mij	(5D)	– " – SDE
1F	0 0 0 1	0 1 0 0	14	Mij	(5C)	– " – SIN
20	0 0 0 1	0 1 0 1	15	Mij	(5B)	– " – SDE
21	0 0 0 1	0 1 0 0	14	Mij	(5A)	– " – SIN
22	0 0 0 1	0 1 0 1	15	Mij	(59)	– " – SDE
23	0 0 0 1	0 1 0 0	14	Mij	(58)	– " – SIN
24	0 0 0 1	0 0 1 0	12	Mij	(57)	– " – LDE
25	0 0 0 1	0 0 0 0	10	Mij	(56)	– " – LIN
26	0 0 0 1	0 0 1 0	12	Mij	(55)	– " – LDE
27	0 0 0 1	0 0 0 0	10	Mij	(54)	– " – LIN
28	0 0 0 1	0 0 1 0	12	Mij	(53)	– " – LDE

Address	Mi: 7 6 5 4	3 2 1 0	Mi hex	Mi type	Label	Operation
29	0 0 0 1	0 0 0 0	10	Mij	(52)	Jump according to CONROM (LIN)
2A	0 0 0 1	0 0 1 0	12	Mij	(51)	– " – LDE
2B	1 1 0 0	1 0 1 0	CA	Mit	LIN(50)	Reg → MARM, select MAR (Mit d5)
2C	0 1 1 0	1 0 1 0	6A	Mie 26		Set reg address: 0 to 1
2D	1 1 0 0	0 0 1 0	C2	Mit		Reg → MARL
2E	1 1 1 0	0 0 0 1	E1	Mit		MEM → AR, select IC MEMA
2F	0 0 1 0	0 1 0 1	25	Mia		Push stack
30	0 1 0 0	0 1 1 1	47	Mie 7		Decrement stack counter
31	1 1 1 0	0 0 1 0	E2	Mit	INC	Reg → AR
32	0 0 1 0	0 1 1 0	26	Mia		AR+1 → Reg
33	0 0 0 0	1 0 0 0	08	Mis		Skip if Zero = 1
34	1 1 1 1	0 0 1 1	F3	Mit		Stack → AR, end
35	0 1 1 0	1 0 0 1	69	Mie 25		Set reg address: 0 to 0
36	1 1 1 0	0 0 1 0	E2	Mit		Reg → AR
37	0 0 1 0	0 1 1 0	26	Mia		AR+1 → Reg
38	1 1 1 1	0 0 1 1	F3	Mit		Stack → AR, end
39	1 1 0 0	0 0 1 0	C2	Mit	LDE	Reg → MARL
3A	0 1 1 0	1 0 0 1	69	Mie 25		Set reg address: 0 to 0
3B	1 1 0 0	1 0 1 0	CA	Mit		Reg → MARM, select MAR (Mit d5)
3C	0 1 1 0	1 0 1 0	6A	Mie 26		Set reg address: 0 to 1
3D	1 1 1 0	0 0 0 1	E1	Mit		MEM → AR, select IC MEMA
3E	0 0 1 0	0 1 0 1	25	Mia		Push stack
3F	0 1 0 0	0 1 1 1	47	Mie 7		Decrement stack counter
40	0 0 1 0	1 0 0 0	28	Mia	DEC	Reg → AR (set Z)
41	0 0 0 0	1 0 0 0	08	Mis		Skip if Zero = 1
42	0 0 0 0	0 1 0 1	05	Mir		Increment address by 2 x 4 + 1 ($\bar{Z}$)
43	0 0 1 0	0 1 1 1	27	Mia		AR–1 → Reg
44	0 1 1 0	1 0 0 1	69	Mie 25		Set reg address: 0 to 0
45	1 1 1 0	0 0 1 0	E2	Mit		Reg → AR
46	0 0 1 0	0 1 1 1	27	Mia		AR–1 → Reg
47	0 0 0 0	1 0 0 0	08	Mis		Skip if zero = 1
48	1 1 1 1	0 0 1 1	F3	Mit		Stack → AR, end
49	0 0 1 0	0 0 0 1	21	Mia		Set zero according to AR
4A	1 1 1 1	0 0 1 1	F3	Mit		Stack → AR, end
4B	0 0 1 0	0 1 1 1	27	Mia	$\bar{Z}$	AR–1 → Reg
4C	0 0 0 0	1 0 0 0	08	Mis		Skip if zero = 1
4D	1 1 1 1	0 0 1 1	F3	Mit		Stack → AR, end
4E	0 1 1 0	1 0 0 1	69	Mie 25		Set reg address: 0 to 0
4F	0 0 1 0	1 0 0 0	28	Mia		Reg → AR (set Z)
50	1 1 1 1	0 0 1 1	F3	Mit		Stack → AR, end
51	0 1 1 0	1 0 0 0	68	Mie 24	SIN	Set reg address: 3 to 0
52	1 1 0 0	1 0 1 0	CA	Mit		Reg → MARM, select MAR (Mit d5)
53	0 1 1 0	1 0 1 0	6A	Mie 26		Set reg address: 0 to 1
54	1 1 0 0	0 0 1 0	C2	Mit		Reg → MARL
55	0 1 0 0	1 1 1 1	4F	Mie 15		Set write MEM
56	0 0 1 0	0 1 0 1	25	Mia		Push stack
57	0 1 0 0	0 1 1 1	47	Mie 7		Decrement stack counter
58	0 1 1 0	0 0 1 1	63	Mie 19		Reset write MEM
59	0 1 0 0	1 1 0 0	4C	Mie 12		Select IC MEMA

Address	Mi: 7 6 5 4	3 2 1 0	Mi hex	Mi type	Label	Operation
5A	0 0 0 1	0 0 0 1	11	Mij		Jump according to CONROM (INC)
5B	0 1 1 0	1 0 0 0	68	Mie 24	SDE	Set reg address: 3 to 0
5C	1 1 0 0	0 0 1 0	C2	Mit		Reg → MARL
5D	0 1 1 0	1 0 0 1	69	Mie 25		Set reg address: 0 to 0
5E	1 1 0 0	1 0 1 0	CA	Mit		Reg → MARM, select MAR (Mit d5)
5F	0 1 0 0	1 1 1 1	4F	Mie 15		Set write MEM
60	0 0 1 0	0 1 0 1	25	Mia		Push stack
61	0 1 0 0	0 1 1 1	47	Mie 7		Decrement stack counter
62	0 1 1 0	0 0 1 1	63	Mie 19		Reset write MEM
63	0 1 0 0	1 1 0 0	4C	Mie 12		Select IC MEMA
64	0 1 1 0	0 1 0 0	6A	Mie 26		Set reg address: 0 to 1
65	0 0 0 1	0 0 1 1	13	Mij		Jump according to CONROM (DEC)
66	0 0 0 1	1 1 1 0	1E	Mij	OP4	Jump by adding $\overline{arg}$+1
67	0 0 0 0	0 0 1 1	03	Mir	(4F)	Increment address by 4x4+1 (RET)
68	0 0 0 0	0 0 1 0	02	Mir	(4E)	– " – 5x4+1 (CHPR)
69	0 1 1 1	0 0 0 0	70	Mie 16	(4D)	Disable interrupt, end
6A	0 1 1 1	0 0 0 1	71	Mie 17	(4C)	Enable interrupt, end
6B	0 1 0 1	0 1 0 0	54	Mie 4	(4B)	Set change IC:12, end
6C	0 0 1 1	1 1 1 1	3F	Mia	(4A)	ICL+AR→ICL (acc to OP), end
6D	0 0 0 0	0 0 0 0	00	Mir	(49)	Increment address by 7x4+1 (addr 8A)
6E	0 0 0 1	1 0 0 0	18	Mij	(48)	Jump according to CONROM (DRCM)
6F	0 0 0 0	0 0 0 1	01	Mir	(47)	Increment address by 6x4+1 (POP)
70	0 0 1 1	0 1 0 1	35	Mia	(46)	Push stack, end
71	0 0 0 0	0 0 1 0	02	Mir	(45)	Increment address by 5x4+1 (SR)
72	0 0 0 0	0 0 0 1	01	Mir	(44)	– " – 6x4+1 (SL)
73	0 0 0 1	0 1 1 1	17	Mij	(43)	Jump according to CONROM (TR)
74	0 0 0 0	0 0 0 1	01	Mir	(42)	Increment address by 6x4+1 (FT)
75	0 1 0 1	0 1 1 0	56	Mie 6	(41)	Select reg block 1, end
76	0 1 0 1	0 1 0 1	55	Mie 5	(40)	– " – 0, end
77						
78	0 1 0 0	0 1 1 1	47	Mie 7	RET	Decrement stack counter
79	1 0 1 0	0 0 1 1	A3	Mit		Stack → ICM
7A	0 1 0 0	0 1 1 1	47	Mie 7		Decrement stack counter
7B	1 0 0 0	1 0 1 1	8B	Mit		Stack → ICL
7C	0 1 0 1	0 0 1 0	52	Mie 2		Increment IC, end
7D	0 1 0 0	0 1 1 1	47	Mie 7	CHPR	Decrement stack counter
7E	1 1 1 0	0 1 1 1	E3	Mit		Stack → AR
7F	0 1 0 0	0 1 1 1	47	Mie 7		Decrement stack counter
80	1 0 1 0	1 0 1 1	AB	Mit		Stack → RCZ ICM
81	0 1 0 0	0 1 1 1	47	Mie 7		Decrement stack counter
82	1 0 0 0	1 0 1 1	8B	Mit		Stack → ICL
83	0 1 1 0	0 0 0 1	61	Mie 17		Enable interrupt
84	0 1 1 1	0 0 1 0	72	Mie 18		Reset priority logic, end
85						
86	0 0 1 0	0 0 1 1	23	Mia	SR	AR shift right
87	0 0 1 1	0 0 0 1	31	Mia		Set zero according to AR, end
88	0 1 0 0	0 1 1 1	47	Mie 7	POP	Decrement stack counter
89	1 1 1 1	0 0 1 1	F3	Mit		Stack → AR, end
8A	0 1 0 1	1 0 0 1	59	Mie 9		Reset, end

Address	Mi: 7 6 5 4	3 2 1 0	Mi hex	Mi type	Label	Operation
8B	0 0 1 0	0 0 1 0	22	Mia	SL	AR shift left
8C	0 0 1 1	0 0 0 1	31	Mia		Set zero according to AR, end
8D	0 0 1 0	0 1 0 1	25	Mia	FT	Push stack
8E	0 1 0 0	0 1 1 1	47	Mie 7		Decrement stack counter
8F	0 1 0 0	0 0 0 1	41	Mie 1		Set IR, increment IC
90	0 0 0 1	1 1 1 0	1E	Mij		Jump by adding $\overline{arg}$+1
91					(F)	
92					(E)	
93					(D)	
94					(C)	
95	0 0 0 1	1 0 1 1	1B	Mij	(B)	Jump according to CONROM (ASC)
96					(A)	
97	0 0 0 1	1 0 1 1	1B	Mij	(9)	Jump according to CONROM (ASC)
98	0 0 0 1	1 0 1 1	1B	Mij	(8)	– " – (ASC)
99					(7)	
9A	0 0 0 1	0 1 1 0	16	Mij	(6)	– " – (LD)
9B	0 0 0 0	0 1 0 0	04	Mir	(5)	Increment address counter by 3x4+1 (DD)
9C	0 0 0 0	0 0 1 1	03	Mir	(4)	– " – 4x4+1 (ID)
9D					(3)	
9E	0 0 0 0	0 1 0 0	04	Mir	(2)	– " – 3x4+1 (LS)
9F	0 0 0 0	0 1 1 0	06	Mir	(1)	– " – 1x4+1 (DS)
A0	0 1 0 0	1 0 0 0	48	Mie 8	(IS)0	Change OP → arg
A1	1 1 1 0	0 0 1 0	E2	Mit		Reg → AR
A2	0 0 1 0	0 1 1 0	26	Mia		AR+1 → Reg
A3	1 1 1 1	0 0 1 1	F3	Mit		Stack → AR, end
A4	0 1 0 0	1 0 0 0	48	Mie 8	DS	Change OP → arg
A5	1 1 1 0	0 0 1 0	E2	Mit		Reg → AR
A6	0 0 1 0	0 1 1 1	27	Mia		AR–1 → Reg
A7	1 1 1 1	0 0 1 1	F3	Mit		Stack → AR, end
A8	0 1 0 0	1 0 0 0	48	Mie 8	DD	Change OP → arg
A9	0 1 1 0	1 0 1 0	6A	Mie 26		Set reg address: 0 to 1
AA	0 0 0 1	0 0 1 1	13	Mij		Jump according to CONROM (DEC)
AB	0 1 0 0	1 0 0 0	48	Mie 8	LS	Change OP → arg
AC	0 0 0 0	0 1 0 1	05	Mir		Increment address counter by 2x4+1 (LB2)
AD	0 1 0 0	1 0 0 0	48	Mie 8	ID	Change OP → arg
AE	0 1 1 0	1 0 1 0	6A	Mie 26		Set reg address: 0 to 1
AF	0 0 0 1	0 0 0 1	11	Mij		Jump according to CONROM (INC)
B0	0 1 0 0	1 0 0 0	48	Mie 8	LD	Change OP → arg
B1	1 1 1 0	0 0 0 1	E1	Mit		Memory → AR
B2	0 1 0 0	0 0 1 0	42	Mie 2		Increment IC
B3	0 0 1 0	0 0 0 0	20	Mia		AR → Reg
B4	0 1 1 0	1 0 1 0	6A	Mie 26		Set reg address: 0 to 1
B5	1 1 1 0	0 0 0 1	E1	Mit	LB2	Memory → AR
B6	0 1 0 0	0 0 1 0	42	Mie 2		Increment IC
B7	0 0 1 0	0 0 0 0	20	Mia		AR → Reg
B8	1 1 1 1	0 0 1 1	F3	Mit		Stack → AR, end
B9	0 0 1 0	0 1 0 1	25	Mia	TR	Push stack

Address	Mi: 7 6 5 4	3 2 1 0	Mi hex	Mi type	Label	Operation
BA	0 1 0 0	0 1 1 1	47	Mie 7		Decrement stack counter
BB	0 1 0 0	0 0 0 1	41	Mie 1		Set IR, increment IC
BC	0 0 1 0	1 0 0 0	28	Mia		Reg → AR
BD	0 1 0 0	1 0 0 0	48	Mia 8		Change OP → arg
BE	0 0 1 0	0 0 0 0	20	Mia		AR → Reg
BF	1 1 1 1	0 0 1 1	F3	Mit		Stack → AR, end
C0	0 0 1 0	0 1 0 1	25	Mia	DRCM	Push stack
C1	0 1 0 0	0 1 1 1	47	Mie 7		Decrement stack counter
C2	0 1 0 0	0 0 0 1	41	Mie 1		Set IR, increment IC
C3	0 1 0 0	1 0 0 0	48	Mie 8		Change OP → arg
C4	0 1 1 0	1 0 1 0	6A	Mie 26		Set reg address: 0 to 1
C5	1 1 1 0	0 0 1 0	E2	Mit		Reg → AR
C6	0 1 0 0	0 0 0 0	40	Mie 0		Change arg → OP
C7	0 0 1 0	1 1 1 0	2E	Mia		Compare AR and reg
C8	0 1 0 0	1 0 0 0	48	Mie 8		Change OP → arg
C9	0 1 1 0	1 0 0 1	69	Mie 25		Set reg address: 0 to 0
CA	1 1 1 0	0 0 1 0	E2	Mit		Reg → AR
CB	0 1 0 0	0 0 0 0	40	Mie 0		Change arg → OP
CC	0 0 0 0	1 0 0 0	08	Mis		Skip if zero = 1
CD	0 0 0 1	1 0 1 0	1A	Mij		Jump according to CONROM (Z=0)
CE	0 0 1 0	1 1 1 0	2E	Mia		Compare AR and reg
CF	1 1 1 1	0 0 1 1	F3	Mit		Stack → AR, end
D0	0 0 0 0	1 1 0 0	0C	Mis	Z = 0	Skip if carry = 1
D1	0 0 0 0	0 1 1 0	06	Mir		Increment address counter by 1x4+1 (C=0)
D2	0 0 1 0	1 1 1 0	2E	Mia		Compare AR and reg
D3	0 1 1 0	1 0 1 1	6B	Mie 27		Reset zero
D4	1 1 1 1	0 0 1 1	F3	Mit		Stack → AR, end
D5						
D6	0 0 1 0	1 1 1 0	2E	Mia	C = 0	Compare AR and reg
D7	0 0 0 0	1 0 0 0	08	Mis		Skip if zero = 1
D8	1 1 1 1	0 0 1 1	F3	Mit		Stack → AR, end
D9	0 1 1 0	1 1 0 0	6C	Mie 28		Reset carry
DA	0 1 1 0	1 0 1 1	6B	Mie 27		Reset zero
DB	1 1 1 1	0 0 1 1	F3	Mit		Stack → AR, end
DC	0 1 1 0	0 1 0 0	64	Mie 20	OP7	Set I/O data to MPU
DD	0 0 0 0	1 0 0 1	09	Mis		Skip if arg = F
DE	0 0 0 0	1 1 1 0	0E	Mis		Skip if Mark (carry) = 0
DF	0 1 0 0	0 0 0 1	41	Mie 1		Set IR, increment IC
E0	0 1 1 0	1 1 1 1	6F	Mie 31		No op
E1	1 1 1 0	1 0 0 0	E8	Mit		I/O data → AR, carry, zero
E2	0 1 1 1	1 1 1 1	7F	Mie 31		No op, end
E3	0 1 1 0	0 1 0 1	65	Mie 21	OP6	Set enable I/O data
E4	0 1 1 0	0 1 1 0	66	Mie 22		Set I/O data from MPU
E5	0 0 0 0	1 0 0 1	09	Mis		Skip if arg = F
E6	0 0 0 0	1 1 1 0	0E	Mis		Skip if Mark (carry) = 0
E7	0 1 0 0	0 0 0 1	41	Mie 1		Set IR, increment IC
E8	0 1 1 0	1 1 1 1	6F	Mie 31		No op
E9	0 1 1 0	0 1 1 1	67	Mie 23		Reset I/O data from MPU

Address	Mi: 7 6 5 4	3 2 1 0	Mi hex	Mi type	Label	Operation
EA	0 1 1 0	1 1 1 1	6F	Mie 31		No op
EB	0 1 1 1	1 1 1 1	7F	Mie 31		No op, end
EC	0 1 0 0	1 0 0 0	48	Mie 8	ASC	Change OP → arg
ED	0 1 1 0	1 1 0 1	6D	Mie 29		Set Mark (carry) to 1
EE	0 0 0 1	1 1 1 1	1F	Mij		Jump according to OP
EF	0 0 0 0	1 0 0 1	09	Mis	OP3	Skip if arg = F
F0	0 0 0 1	1 1 0 0	1C	Mij		Jump according to CONROM (JCF)
F1	1 0 0 0	0 1 0 0	84	Mit		ICL → stack
F2	1 0 0 0	0 1 0 1	85	Mit		RCZ ICM → stack
F3	0 0 1 1	0 1 0 0	34	Mia		IC + MEM → IC, end

Note! Unprogrammed outputs (empty positions in the table) can be low (Intersil) or high (MMI and other manufacturers).
The table is valid for any manufacturer except in addresses 6D and 8A where the following is valid

Intersil (in the table)

6D	0 0 0 0	0 0 0 0	00	Mir	Inc address
8A	0 1 0 1	1 0 0 1	59	Mie 9	Reset, end

MMI and other manufacturers

6D	0 1 0 1	1 0 0 1	59	Mie 9	Reset, end
8A	unprogrammed (ie 1)				

Memory

Contents

GENERAL 1

MEMORY 4/4 3

Brief outline 5
Detailed description 6

FIGURES

1. Memory arrangement in cluster configuration 2
2. Block diagram of the MEM 4
3. Memory address arrangement 4
4. Table showing address modification 6
5. Table for data selection 7

GENERAL

All control units used in the Alfaskop 3500 can accommodate two memory boards. The entire instruction memory (INM) and constant memory (COM), which comprise PROMs (programmable read only memories) are contained on the memory boards. In addition, there is a read/write memory (RWM) which can be utilized as a working memory (WOM), display memory (DIM) and printer memory (PRM) on the memory boards. In cluster control units, DIM, PRM and even parts of WOM are contained in display adapters (DA) or printer adapters (PA). To facilitate the adaptation of memory capacity to requirements, the following memory board variants are available.

No	Max capacity	Remarks
E07150 0020	4K ROM, 4K RWM	2K RWM always mounted, DIP holders for 2K
0021	4K ROM	
0022	4K ROM, 4K RWM	DIP holders for 4K RWM
0130	6K ROM, 2K RWM	1K RWM always mounted, DIP holders for 1K
0131	4K ROM	Address commences at 1000

ROM DIPs are always mounted in holders and can thus be replaced.
The first three types can be inserted at both memory locations while the 6/2 memory can only be inserted at position MEM1 and the 4/0 (0131) memory can only be inserted at position MEM2.

The following options (maximum capacity) can be obtained by selecting different memory boards.

A One memory board 4/4

Address	0000 – 0FFF	ROM	(4K bytes)
	1000 – 1FFF	RWM	(4K bytes)

B Two memory boards 4/4 (or one 4/4 + one 4/0)

Address	0000 – 0FFF	ROM	(4K bytes)	Pos MEM1
	1000 – 1FFF	ROM	(4K bytes)	Pos MEM2
	2000 – 2FFF	RWM	(4K bytes)	Pos MEM1
	3000 – 3FFF	RWM	(4K bytes)	Pos MEM2

C One memory board 6/2

Address	0000 – 17FF	ROM	(6K bytes)
	1800 – 1FFF	RWM	(2K bytes)

or

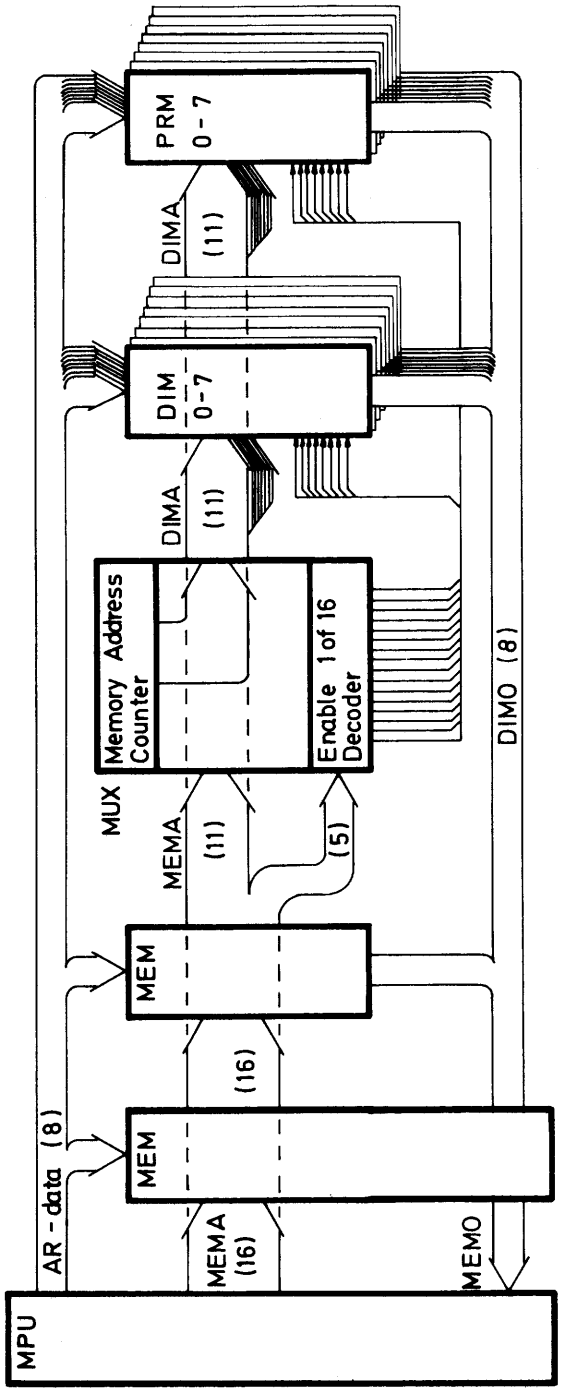


Fig. 1. Memory arrangement in cluster configuration

Address	0000 – 17FF	ROM	(6K bytes)
	2000 – 27FF	RWM	(2K bytes)

The following addresses are used for memories in adapters in cluster control units.

4000 – 47FF	Display area for display unit	1	(2K bytes)
4800 – 4FFF		2	
5000 – 57FF		3	
5800 – 5FFF		4	
6000 – 67FF		5	
6800 – 6FFF		6	
7000 – 77FF		7	
7800 – 7FFF		8	
8000 – 87FF	Print area for data printer	1	(2K bytes)
8800 – 8FFF		2	
9000 – 97FF		3	
9800 – 9FFF		4	
A000 – A7FF		5	
A800 – AFFF		6	
B000 – B7FF		7	
B800 – BFFF		8	

Each display area and print area can contain 2K bytes or 1K byte.

Addresses C000 – EFFF are used for the RWM memory in the channel communication adapter (CCA), and this memory normally contains only 2K.
Addresses FFF0 – FFFA are used to address the strappable constants on the memory at position MEM1.

MEM

Fig. 1 shows how the memory is arranged in a cluster control unit. This is described in greater detail in the section entitled "Cluster operation".

Only MEM 4/4 (0020) is described in the following. The other variants function in the same way. The only difference is that variants 0130 and 0131 have no address modifier.

MEMORY 4/4

This memory can be inserted at two locations in the units, and it is automatically provided with different addresses for its memory cells (viewed eternally). One 8-bit byte is stored in each cell. This memory can be expanded in steps to a maximum of 4K bytes of ROM and 4K bytes of RWM. For ROM, the expansion steps comprise 256 bytes. For RWM, the expansion steps comprise 1K bytes. Also included is a constant memory having 10 strappable bytes and one byte that is selected with switches.

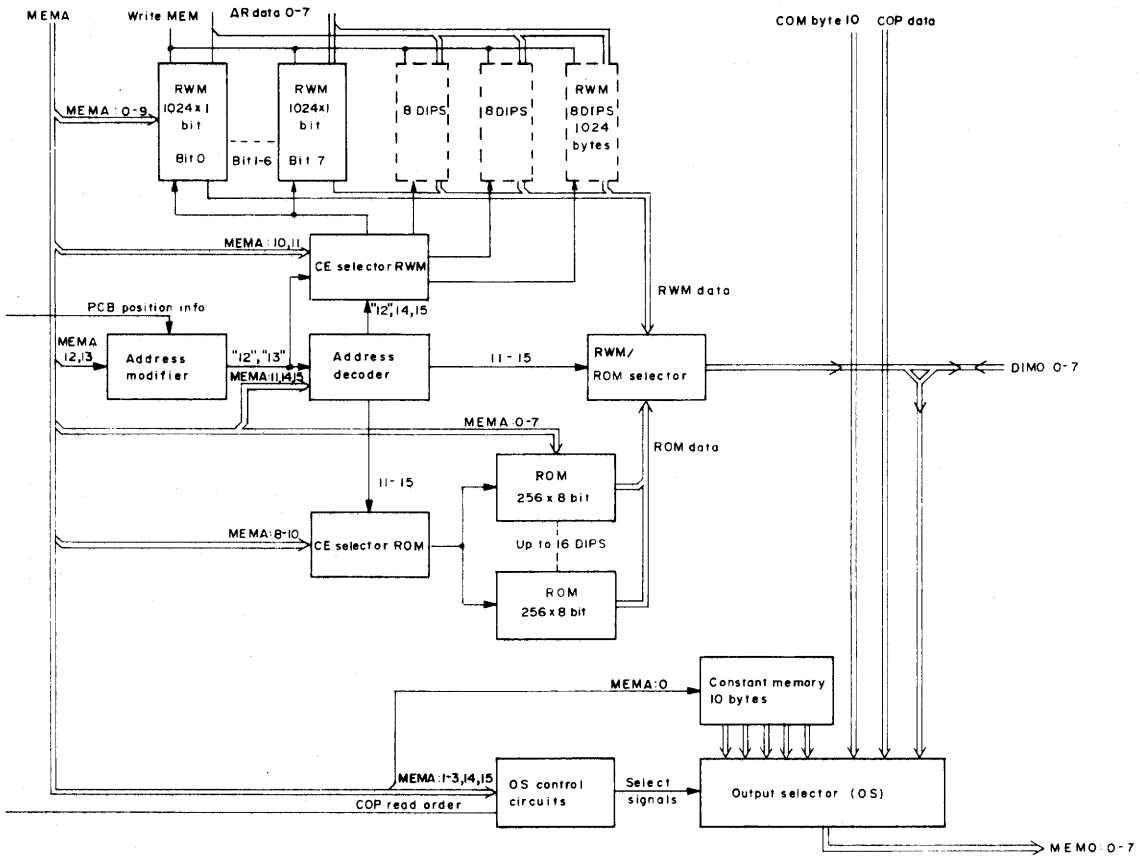


Fig. 2. Block diagram of the MEM

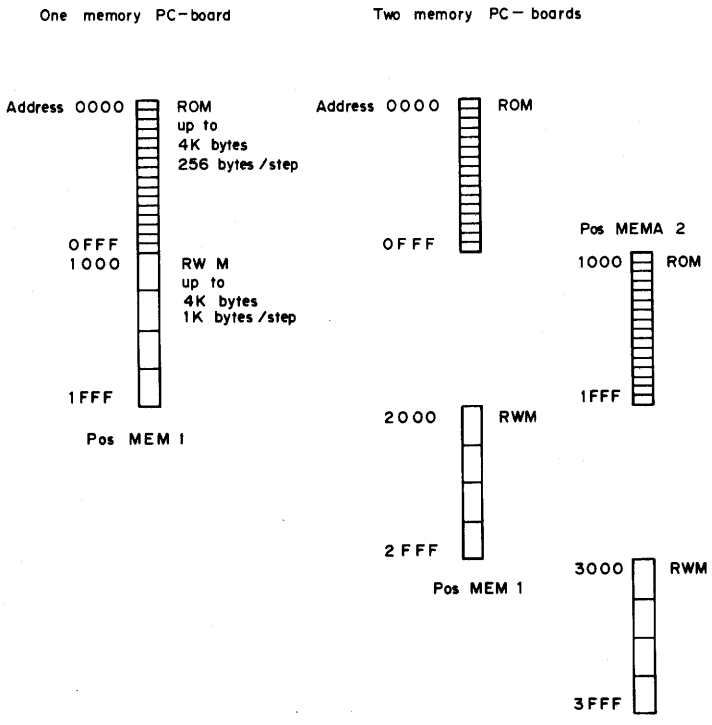


Fig. 3. Memory address arrangement

Brief outline

The block diagram in Fig. 2 presents an overview of how the memory functions. The address (MEMA) is sent to the address inputs to the memory DIPs and to the decoders. The read/write memory (RWM) is organized in blocks of 1024 bytes. The 10 least significant address bits are entered into all blocks, thus addressing one cell in each block. The read-only memory (ROM) is organized in blocks of 256 bytes. As a result, the eight least significant bits are entered into the ROM blocks.

Memory blocks are selected in several stages. It must be possible to use the memory at two position in the board mounting frame. If only one memory is used, the internal address is the same as the external address. See Fig. 3. This memory shall be inserted at the position marked MEM1. If another memory is inserted at the position marked MEM2, the addresses for the RWM part are moved on the board at position MEM1. The addresses for the ROM and RWM parts of the board found at location MEM2 are shown in Fig. 3. This arrangement permits the ROM to be addressed sequentially at all times.

The address modifier senses the strapping on the interconnection board and changes bits 12 and 13 from external to internal address bits.

The addresses of the constant memory are FFF0 – FFF9 (strapping bytes) and FFFA (switch byte). Only the strapping plugs on the memory board inserted at position MEM1 are utilized.

The address decoder senses bits 14 and 15, thus ascertaining whether the RWM part or ROM part is addressed (both bits shall be low). Internal bits 12 and 13 specify whether a) the RWM part or ROM part of the first board or b) the second board is addressed. Control signals from the address decoder are sent to the two CE selectors so that the one used for RWM operates when RWM is addressed and the one used for ROM operates when ROM is addressed. In addition, signals are sent to the RWM/ROM selector which is set in compliance with the signals sent to the CE selectors.

CE selector RWM selects, by means of address bits 10 and 11, the 1K block of eight DIPs in question. All these DIPs receive enable signals. CE selector ROM selects, by means of address bits 8–10 and control signals received from the address decoder, one of the 16 ROM DIPs. The selected DIP receives an enable signal.

Data from the RWM/ROM selector is sent out to the display logic. The two memory boards are joined together on the interconnection board, but data arrives, of course, from only one of the memory boards at any given point in time. From the RWM/ROM selector, or from the second memory board, data is also sent to OS (output selector) which selects data for the output to MPU (MEMO). OS can select data from eight different sources (RWM/ROM selector, COP data from the control panel, COM byte 10 from a group of switches and five bytes from the constant memory). The constant memory provides different information, depending upon the least significant address bit. This means that ten constant bytes are available. The control signals (addresses) sent to OS come from the OS control circuits.

Both the ROM and RWM memories are static and thus require no clock pulses. They do not have to be refreshed. The access time is about 1 μs for both parts of the memory. The delays encountered in the decoders at the memory CE inputs have no effect. Data at the outputs remains stable until the address sent to the memory board is changed. This data varies only during the access cycle.

When information is to be entered into the RWM part of the memory, MPU first puts out the data

(AR data), then the address and finally a write pulse that endures for about 825 ns (Write MEM low). When the write pulse vanishes, the data has been recorded. The address data and AR data are not changed before a time interval of about 160 ns has elapsed. This means that the memory elements are provided with ample time for recovery and data-hold.

Detailed description

The address modifier senses whether the ROM or RWM part of the memory is addressed. If ROM is addressed, outputs IC65/8 and IC65/6 shall both be high. If RWM is addressed, IC65/6 shall be low and IC65/8 shall be high. The following table illustrates how this is accomplished by means of incoming address bits and cabling.

		MEMA		IC79				IC65/8	IC65/6	Selected part
		13	12	A1	A0	XB	XC			
One MEM PC-board		0	0	X	X	X	X	H	H	ROM
at pos MEM1		0	1	X	X	X	X	H	L	RWM
P2/62 +		1	0	X	X	X	X	L	H	–
P2/63 +		1	1	X	X	X	X	L	L	–
Two MEM PC-boards	at pos MEM1	0	0	H	H	L	L	H	H	ROM
	P2/62 +	0	1	H	L	H	L	L	H	–
	P2/63 0 V	1	0	L	H	L	H	H	L	RWM
		1	1	L	L	H	H	L	L	–
	at pos MEM2	0	0	H	H	L	L	L	H	–
	P2/62 0 V	0	1	H	L	H	L	H	H	ROM
	P2/63 0 V	1	0	L	H	L	H	L	L	–
		1	1	L	H	H	H	H	L	RWM

Fig. 4. Table showing address modification

If the RWM part is addressed, IC41/1 is high. Moreover, IC77/3 and IC78/12 are low. This means that selectors IC53 and IC40 permit data from RWM to pass through. Address bits 10 and 11 select, via IC54, the particular group of eight RWM DIPs that is to be enabled. If the ROM part is addressed, IC77/3 is high, i.e. selectors IC53 and IC40 permit data from ROM to pass through. If bit 11 is high, IC51/6 – 38/12 go low, and as a result all outputs from IC38 are high. On the other hand, one of the outputs from IC37 is low because of bits 8, 9 and 10. Consequently bits 8–11 can be used to select one of the sixteen possible ROM DIPs.

None of the ROM or RWM DIPs on the board is selected if IC65 provides combinations at its outputs other than those mentioned above. For these other combinations, IC39/3 also goes high, and as a result selectors IC53 and IC40 acquire high-impedance outputs (disconnected). The same thing applies if IC39/8 goes low. This occurs if either of address bits 14 or 15 become one (0 V at input to IC39/9 or IC39/10).

Ten constant bytes are obtained by connecting straps on four plugs at DIP positions 60–63. The bits in the outgoing data (MEMO) are selected by eight selectors of type 74151. From each plug, five signals are sent to the bit selectors, but since address bit 0 is entered into the plugs, ten different bytes can be obtained. The selectors control selector IC72 (type 74157) in accordance with the table which follows.

MEMA		IC72/1	COP read	MEMA			IC59			Selected data
15	14		order	3	2	1	12	10	8	
1	1	L	H	0	0	0	L	L	L	COM byte 0/1
1	1	L	H	0	0	1	L	L	H	COM byte 2/3
1	1	L	H	0	1	0	L	H	L	COM byte 4/5
1	1	L	H	0	1	1	L	H	H	COM byte 6/7
1	1	L	H	1	0	0	H	L	L	COM byte 8/9
1	1	L	H	1	0	1	H	L	H	COM byte 10
0	X	H	H	X	X	X	H	H	L	Data from ROM,RWM
X	0									
X	X	X	L	X	X	X	H	H	H	COP data

Fig. 5. Table for data selection

Modem Communication Adapter

Contents

GENERAL 1

BRIEF OUTLINE 1

- Receiver unit 1
- Transmitter unit 3
- Monitoring 4

DETAILED DESCRIPTION 5

- Receiver unit 5
- Transmitter unit 7

FIGURES

1. Block diagram of modem communication adapter 2
2. Receiver synchronizing 6
3. Oscillator 6
4. Start of sending from MCA 8
5. Delay of resetting RTS 8
6. Line activity lamp 10

GENERAL

The modem communication adapter which is built up on one PC-board of the small size (half board) is used to adapt modem-communication signals in accordance with the requirements imposed by the CCITT V24 interface. Most of the functions are carried out within two LSI circuits which the manufacturer calls the terminal receiver and terminal transmitter. Integrated line drivers and line receivers are used for interfacing with the modem.

The board is provided with a strapping plug which enables the transmission rate, transmission type, transmission code, etc. to be selected. In general, the following description is limited to synchronous transmission, 7-bit code + odd PC bit, half duplex (or simplex).

BRIEF OUTLINE

A block diagram of the modem communication adapter appears in Fig. 1. The strapping plug, instruction decoder and lamp driver blocks are the only ones which are shared by the transmitter unit and receiver unit.

The instruction decoder is connected to the I/O address bus. If the micro processing unit wishes to transfer data to or from MCA via the I/O data bus, it issues a signal "I/O address MCA group" and simultaneously either I/O data to MPU or I/O data from MPU. For each transmission direction, four instructions can be decoded using I/O address a0 and I/O address a1. The following are utilized:

- I/O data from MPU
  - Write transmitter
  - Set request to send
  - Reset request to send
  - Look for SYN

- I/O data to MPU
  - Read receiver

Receiver unit

A terminal receiver always senses data that arrives on Received data. The selector above the receiver can be set so that receiver signal element timing (RET) is entered into the receiver as a clock pulse at the same time that it is sent to "division-by-one". Synchronization of the individual bits then takes place in the modem, and changeovers from ON to OFF and vice versa at RET indicate the centres of the bits received on Received data. If the modem does not provide RET, the straps shall be connected to provide internal clocking. In such case, the selector is set to the other position, whereupon the terminal receiver itself handles bit synchronization. In order for the receiver to operate synchronously with regard to characters, it must have received the SYN detected signal.

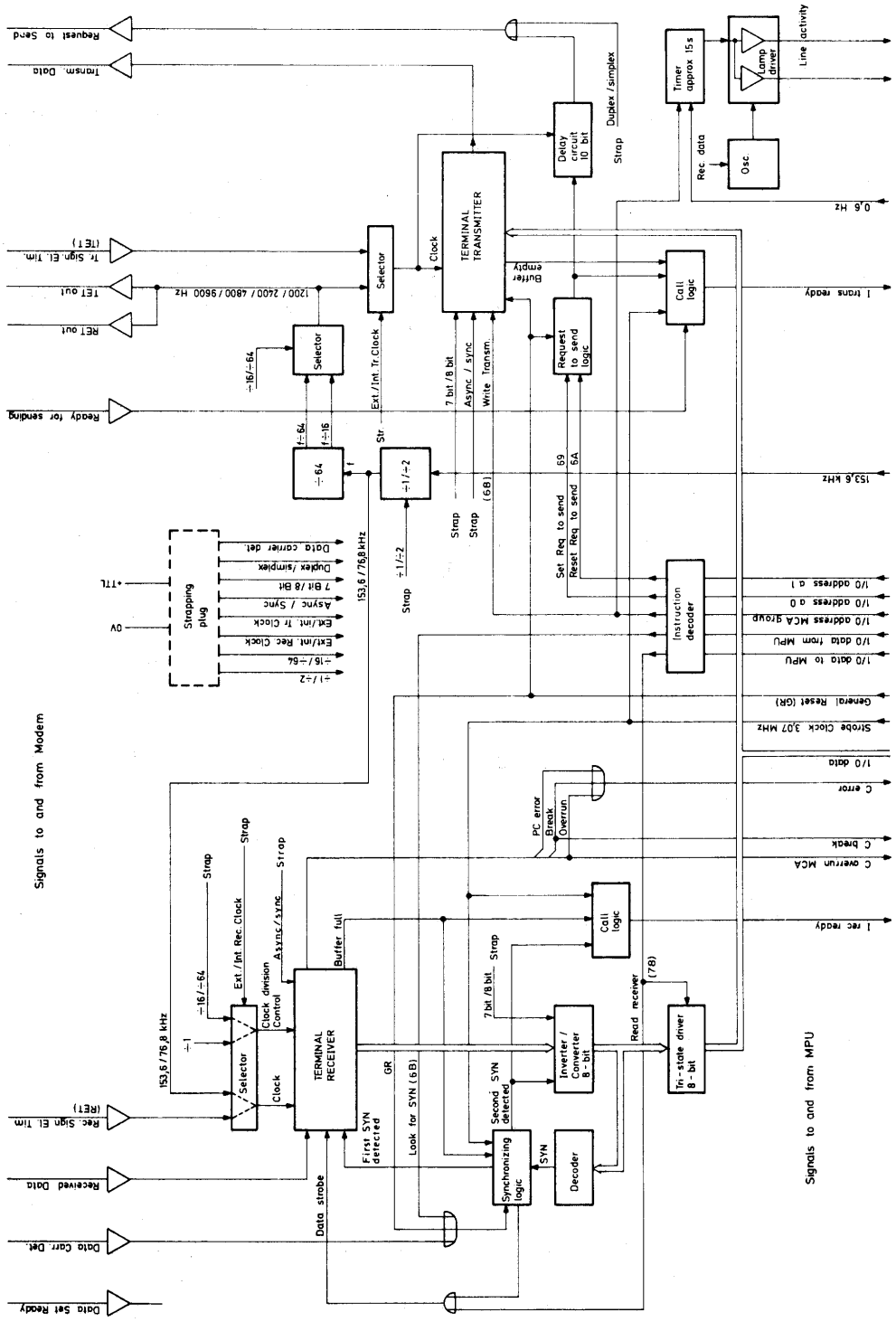


Fig. 1. Block diagram of modem communication adapter

Received data is shifted into a shift register in the receiver from which it is transferred to a buffer register when a character has been received. Before the receiver commences to operate synchronously with regard to characters, it is transparent, i.e. items of shift-register data appear at the output as they are shifted in. In such case, the block inverter/decoder operates as a decoder and sends a signal to the synchronizing logic at times when the character in the receiver is SYN.

As soon as a SYN character is detected, the synchronizing logic sends the SYN detected signal to the receiver. Thereafter, the receiver sends the Buffer full signal each time it receives eight new bits and has entered the corresponding character into the buffer register. The outputs are now open only when the receiver receives the Data strobe signal.

The first character that arrives after SYN detected has been sent to the receiver must also be SYN. The first time the terminal receiver issues the Buffer full signal, the synchronizing logic sends Data strobe to the receiver, whereupon a character is sent out. If the character was SYN, the synchronizing logic issues the second SYN detected to the inverter/decoder which thereafter functions as an inverter. The second SYN detected is also sent to the call logic. For each character that is subsequently received, the call logic sends an interrupt signal to the micro processing unit. If the second character was not SYN, SYN detected vanishes from the receiver input and hunting for SYN commences again.

When the micro processing unit is to fetch a character, it issues a command to the Read receiver via the I/O data bus. The terminal receiver then receives the Data strobe signal, whereupon the received character is sent to the inverter/decoder. Seven of the bits (all but the parity bit) are sent through the tri-state driver which is opened by the Read receiver signal. These seven bits then proceed out onto the I/O data bus (on condition that a 7-bit code is being used for transmission – if a 8-bit code is being used for transmission, all eight bits are entered on to the I/O data bus).

If a parity error is detected in the terminal receiver, MCA sends conditional signal C error to the micro processing unit. This signal is sent out simultaneously with Buffer full, and persists until the next Buffer full arrives. If the receiver has not received any Data strobe signal since the time when it issued the previous Buffer full, it sends out the overrun signal at the same time that the new character is entered into the receiver output register. The old character is thus lost. Conditional signals C error and C overrun MCA are sent to the micro processing unit. These signals persists until the next character has been received. The terminal receiver issues conditional signal Break only in connection with asynchronous transmission (when the stop bit is missing).

Transmitter unit

When the micro processing unit is to transmit via the MODEM, it first sends the Set request to send signal via the I/O address bus. MCA sends Request to send immediately via the request to send logic and delay circuit. The modem equipment is then changed over for sending, and when this is completed, the modem sends the Ready for sending signal. (If a full duplex connection is being used, the micro processing unit need not send Set request to send.) The call logic then sends an interrupt signal to the micro processing unit which enters a character onto the I/O data bus and then sends the instruction Write transmitter via the I/O address bus.

The Terminal transmitter contains, among other things, a buffer register and a shift register which send out data in serial form to the modem. While a character is being shifted out, the next character must be stored in the buffer register so that it can be sent (starting with its first bit) imme-

diately after the last bit in the previous character. As soon as a character has been transmitted from the buffer register to the shift register, the transmitter sends the Buffer empty signal which, via the call logic, causes the interrupt signal to be sent to MPU. If the buffer register is not filled before the next character is to be sent, the transmitter will send eight ones instead.

The parity bit is generated in the terminal transmitter in such a way that an odd number of ones is included in each character that is transmitted. (This means that there will be a parity error if the transmitter is not filled quickly enough.) On the other hand, the micro processing unit must sent out a sufficient number of SYN characters when transmission commences.

After the first character has been entered into the transmitter buffer register, there is a time delay that is approximately equal to the corresponding transmission time for a character before the transmitter transfers the character to the shift register. This delay is imposed because the transmitter commences to meter out eight bit times (while sending ones) after Ready for sending has arrived from the modem.

After the first SYN character has been transferred to the shift register, MPU usually enters the next SYN character into the transmitter immediately. This means that there are normally two characters in the transmitter.

When MPU has entered the last character that it wishes to send in connection with the transmission operation in question, the sending out of the next to the last character has just started. When the last character is transferred to the shift register in the transmitter, MPU receives a new interrupt signal. It responds to this signal by sending the instruction Reset request to send. The Request to send signal which has been sent to the modem must, however, persist while the last character is transmitted, and it must also persist for a brief interval thereafter, so that the modem will actually have time to send the character completely. When the request to send logic is reset, the delay circuit is actuated, thus enabling Request to send to persist throughout a time interval corresponding to ten bits.

Monitoring

The control unit contains a lamp which indicates traffic on the modem line as follows:

- Extinguished when the line is inactive
- Flashes when the modem is only receiving data
- Glows steadily when the control unit is sending data

When the data transmission system operates normally the control unit sends, at least, a reply to a polling call. Normally, there shall be a maximum of a few seconds between transmission operations. The timer includes a counter that is cleared each time the instruction Write transmitter is received. This counter is incremented by a clock signal having a frequency of about 0.6 Hz. If the counter does not have time to count to eight before the next Write transmitter, the lamp driver will keep the lamp glowing, i.e. it glows for about 13 s after the last character is sent out.

If the timer does not provide the lamp-lighting conditions, it is kept flashing by an oscillator. This oscillator oscillates as long as Received data changes between one and zero. If Received data remains either level, the oscillator stops and the lamp is extinguished.

DETAILED DESCRIPTION

In the following circuit description, the boxes on the block diagram are discussed individually.

Receiver unit

- Selector

  While General reset is low, the selector is blocked, resetting the terminal receiver. If Ext RET mode is high (and GR is high) RET is sent via IC3/4 to the receiver clock input. Receiver inputs 27 and 28 go low simultaneously. If Ext RET mode is low and internal clock signal is sent to receiver clock input.

- Terminal receiver
  - Inputs 8 and 9 (Word length selector) are always high, which means that the length of the received characters shall be eight bits (including any parity bit).
  - Input 24 (Status strobe) is always high, and outputs 22, 23, 25 and 26 are thus always open.
  - Inputs 4 (Even parity) and 6 (Asynchronous mode) shall be high for asynchronous transmission and low for synchronous transmission.
  - Inputs 27 and 28 (Count enables) are either both high (Master reset) or both low if RET (obtained from the modem) is to be used as the clock signal. If the period of the internal clock is 64 times the bit time, 28 shall be high. If the period is 16 times the bit time, 27 shall be high.
  - High-to-low transitions at input 1, indicate the centre points of bits at input 3 (Received data). (This applies only when an external clock is used.)
  - The high level at input 20 (Data strobe) opens the data outputs.
  - The high level at input 2 (SYN detected) indicates that the external circuits have detected a SYN character.
  - Output 18 (Internal clock) provides the inverted value of one period of the clock signal at input 1 (at the centre of the bits). For division-by-one, this will be the entire clock signal.
  - Outputs 10, 11, 13–17, 19 provide parallel data output (one = high level).
  - Output 22 (Buffer full) goes high when a received character is transferred to the buffer register, and it remains high until the character is entered onto the outputs by means of a high level at input 20 (Data strobe).
  - Output 25 (Overrun) is high if one or more characters have been transferred to the buffer register without output 20 (Data strobe) having been high.
  - Output 26 (Parity error) is high if the character that is transferred to the buffer register has the wrong parity.
  - Output 23 (Break) is not used for synchronous transmission.

The functioning of the terminal receiver is evident from the block diagram and timing diagram in Fig. 2.

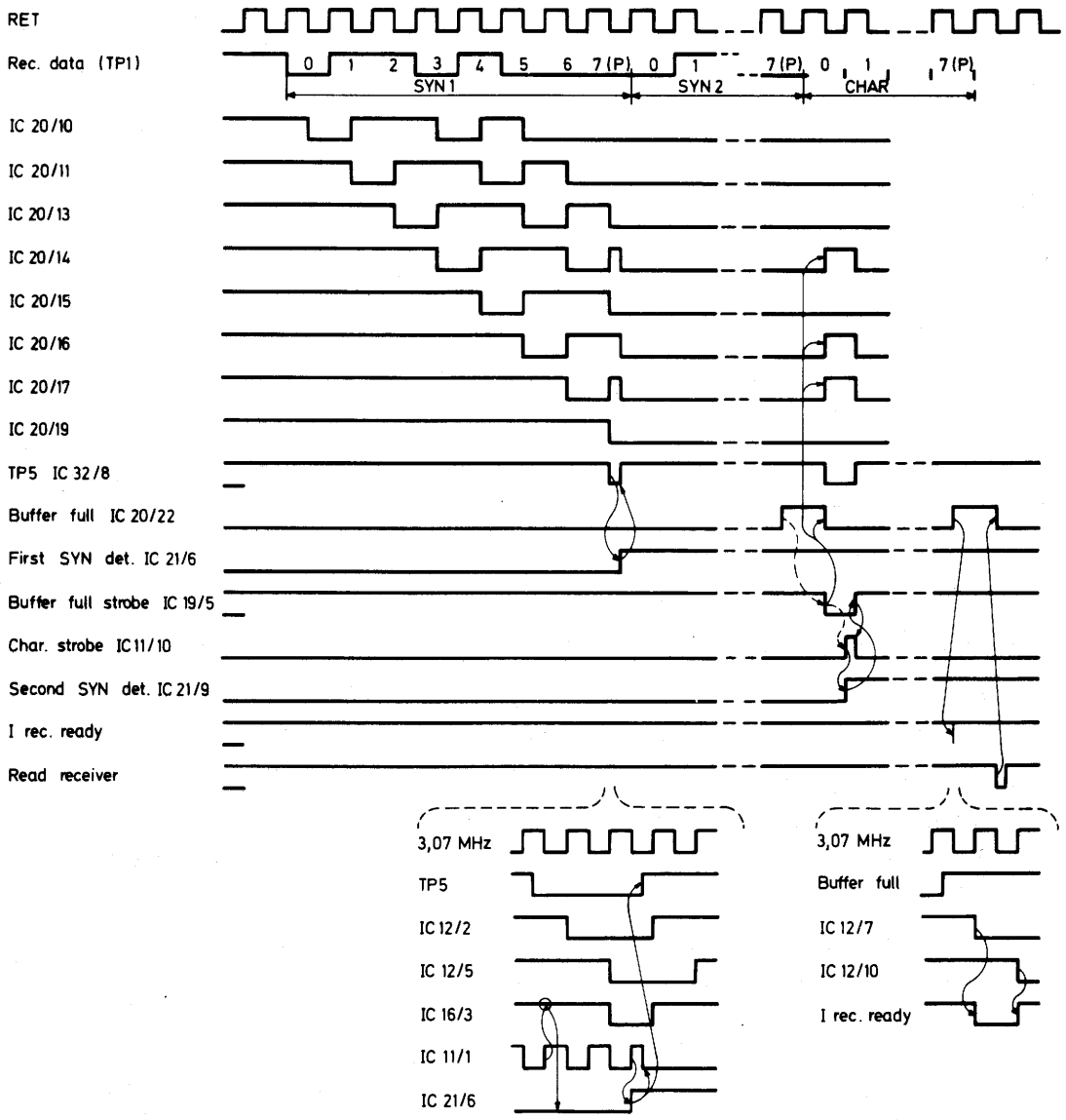


Fig. 2. Receiver synchronizing

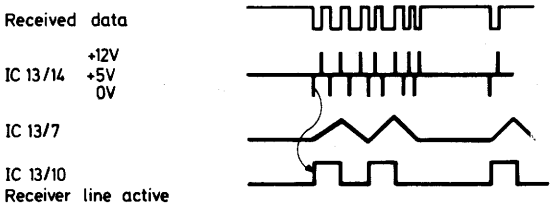


Fig. 3. Oscillator

- Inverter/decoder

  Eight EXCLUSIVE OR gates comprise the lion's share of the inverter/decoder. Gates 25/11, 25/8, 30/8 and 30/3 always function as inverters for data obtained from the terminal receiver since their second inputs are always high. When seven bits (+ parity bit) are expected in the incoming character, 25/5 is also high, i.e. 25/6 functions as an inverter.

  If a SYN character (0001 0110 for 7-bit code) is present at the data outputs of the terminal receiver, all outputs from gates in DIP 25 and 30 will be high.

  After flip-flop 21/9 has been 1-set, all gates function as inverters.

- Synchronizing circuit

  Two consecutive SYN characters are detected by the synchronizing circuits which then send a condition to the call logic and other circuits. This function is illustrated, in general, by the timing diagram in Fig. 2. This timing diagram covers only the events taking place when two SYNs arrive consecutively. The lower part of the diagram shows several cycles drawn to different time scales (frequency of RET is 2.4 kHz at 2 400 bit/s). The following supplementary remarks may be helpful.

  Terminal receiver IC20 indicates at its outputs how incoming data is being shifted through, until IC21/6 (First SYN detected) goes high. After IC21/6 goes high, IC20 provides output data only when its input 20 goes high. Flip-flop 19/5 causes input 20/20 to go high only after the character after the SYN character has been received. If this character is not SYN, input 2 on flip-flop IC21 is high (and input 12 is low) when the pulse from IC11/10 arrives. In such case, hunting for a new SYN commences immediately (terminal receiver becomes transparent again). Hunting also starts if flip-flop IC21 is actuated via input 4.

- Call logic

  After two SYN characters have been detected, each incoming character gives rise to an interrupt signal as shown in the timing diagram in Fig. 2. This applies even if the character is SYN. Output 32/8 (TP5) remains high thereafter, even for SYN.

- Oscillator

  The oscillator provides a square wave for the lamp drive circuit if data arrives on Received data. If Received data does not change, input 14 to inverter IC10 is at +5 V and, in such case, having input 7 close to 0 V is not sufficient to ensure that the inverter will operate (it must be negative). If there is a negative-going transition on Received data, negative spikes (towards 0 V) are obtained at input 14, and the inverter is activated in response to each spike. See Fig. 3. Actually, there will be considerably more crossovers than shown in the diagram. Moreover, the duration of the output pulse is much longer than the distance between crossovers.

MCA

Transmitter unit

- Timing circuits and selector

  If Ext TET mode is high, TET (obtained from the modem) arrives via gate IC4/8, and the other

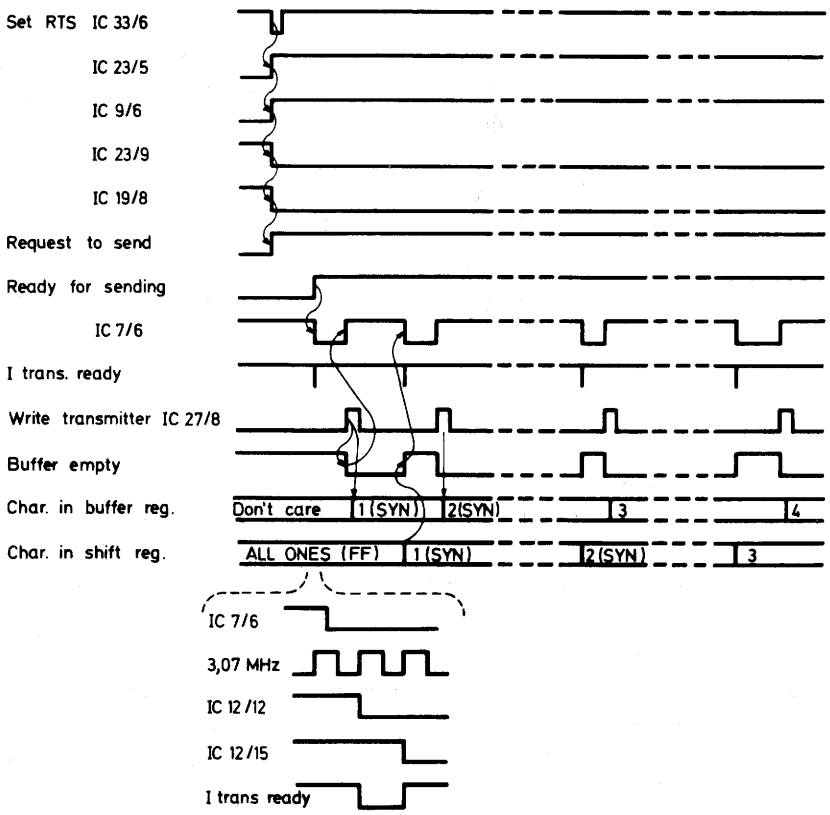


Fig. 4. Start of sending from MCA

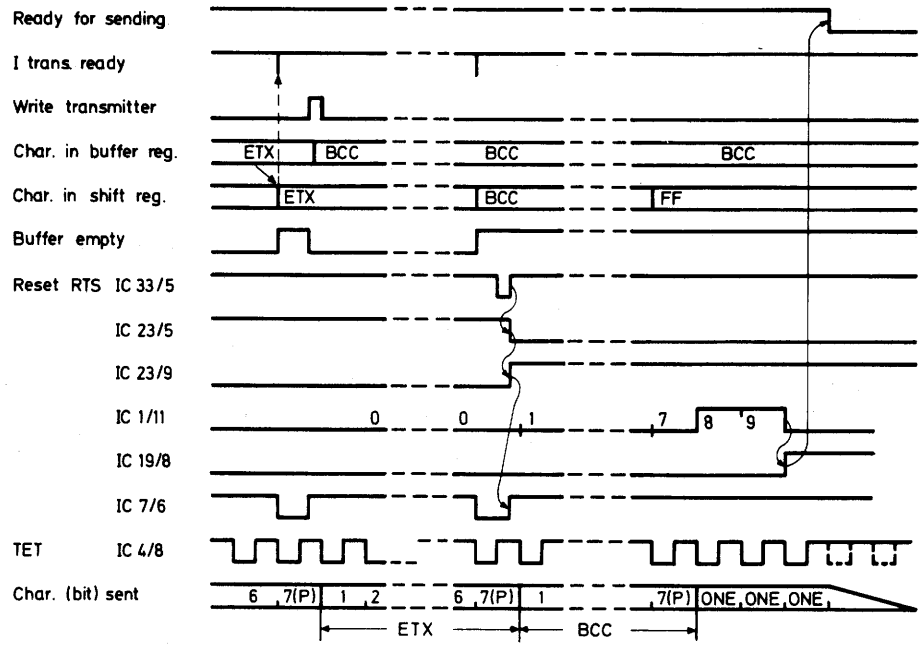


Fig. 5. Delay of resetting RTS

"clock straps" and timing circuits do not perform any functions. If Ext TET mode is low, an internal clock signal arrives which is divided in IC10 and IC5 (by 16 or 64 respectively) via gate IC4/8.

- Terminal transmitter
  - Inputs 11 and 12 (Word length selector) are always high, which means that eight bits (including any parity bit) are transmitted.
  - Input 13 (2nd stop bit select) is always low which provides one stop bit. This input has no effect on synchronous transmission.
  - Inputs 7 and 9 (Count enables) are either both high (Master reset) or both low, which means that the signal obtained from IC4/8 is used directly as a clock signal.
  - Input 17 (Internal parity) is high if the parity bit is to be generated in the terminal transmitter.
  - Input 14 (Asynchronous mode and 16 Even parity) shall be high for asynchronous transmission and low for synchronous transmission.
  - The signal at input 8 (External clock) controls output via output 19 (Data out) so that low--to-high transitions indicate crossovers between two bits.
  - High level at input 21 (Load strobe) transfers data from parallel inputs 1–3, 5, 6, 22–24 and sets output 20 (Buffer empty) low.
  - Input 15 (System ready) has not effect whatsoever in connection with synchronous transmission.
  - Output 10 (Data out) provides serial output data.
  - Output 20 (Buffer empty) goes high when the data in the buffer register is transferred to the shift register and the transmitter can thus receive a new character.

- Request to send logic and call logic

  The timing diagram in Fig. 4 shows how signal Set RTS from the decoder in DIP 33 sends Request to send to the modem. If, just previously, the modem had been in the transmitting mode and had not had time to set Ready for sending low, gate input IC9/4 would be high. In such case, flip-flop IC23/9 is not zero-set before the "old" Ready for sending vanishes. It is also evident from the diagram how the interrupt signal is generated and how the first character is sent. The transmission of the first character from the buffer register does not take place before a time interval corresponding to the transmission time for eight bits has elapsed since Ready for sending went high (see block description).

- Delay circuit

  When transmission is concluded, the modem must not be disconnected before the last character has been sent. In Fig. 5, it has been assumed that the two last characters from the Control unit are ETX and BCC. The sending out of data takes place in the terminal transmitter via a buffer flip-flop so that the output data is displaced one bit after the transmitter shift register. (As a result, the transmitter has one bit time available in which to transmit the next character from the buffer register to the shift register.)

Counter IC1 is released when IC23/9 is zero-set. Thereafter, it is incremented in response to negative-going edges of clock signal TET (note that MPU is generally so fast that incrementation commences one period earlier than shown in Fig, 5). When the counter has counted to ten, the Request to send signal goes low at the modem, whereupon the modem is disconnected. This takes place at different speeds for different modems (similarly, different modems can send or not send TET after disconnection).

- Lamp driver

Counter IC31 is set to 9 by the General reset signal and cleared by the Write transmitter signal. The lamp will light as long as the counter is at 0 through 7. When the counter reaches 8 or 9, it cannot be clocked any further, and during this interval the current sent to the lamp is determined by the Receiver line active signal.

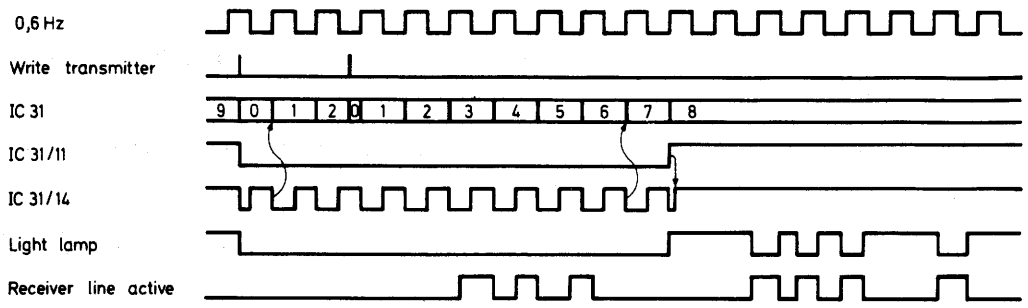


Fig. 6. Line activity lamp

External Bus Adapter

Contents

GENERAL 1

DATA TRANSFER VIA EDB 2
- Control signals on the EDB 2
- EBA operation during transfer 3

DETAILED DESCRIPTION 8
- Transmitting general call 8
- Receiving general call 9
- Data transfer, transmitting unit 10
- Data transfer, receiving unit 11
- Changeover from receiving to transmitting 11
- Changeover from transmitting to receiving 12
- Bus control circuits 12

FIGURES

1. External data bus (EDB) 1
2. Control signal timing 2
3. Block diagram of the external bus adapter (EBA) 4
4. EBA timing diagram 5
5. Changing transfer direction 7
6. Transmitting general call 8
7. Receiving general call 9
8. Data transfer, transmitting unit 10
9. Data transfer, receiving unit 11
10. Changeover from receiving to transmitting 11
11. Changeover from transmitting to receiving 12
12. Bus control 13

GENERAL

The external bus adapter (EBA) can be used in Control Unit 3501/3502, Expansion Module 3531 and Control Unit 3170. It is used for data transfer between microprocessing units (MPU) across an external data bus (EDB). It comprises a PC board of the small size (half board) and is only inserted when the terminal system incorporates more than one micro processing unit.

Each control unit is provided with two connectors for the EDB cable and a master/slave switch. One of the connectors is intended for the cable from the preceeding control unit, the other for the cable to the next control unit. A terminating connector must be inserted in the first and the last control unit. Each line in the EDB is driven from a transistor as shown in Fig. 1. The +5 V supply is taken from the master unit (the one that is connected to modem via MCA) when the master/slave switch is put in position master. The switch must be in position slave in all other control units.

Fig. 1 shows the signals available on the EDB. In addition to the wires for the signals, there are wires for +5 V and 0 V. The two status bits are normally not used by the program and are in such case not transferred. The nine data bits contain an odd number of ones during transfer. Only the master can send the reset slave signal, all units can send all other signals. Logical one is low level (0 V) for all signals but data accepted.

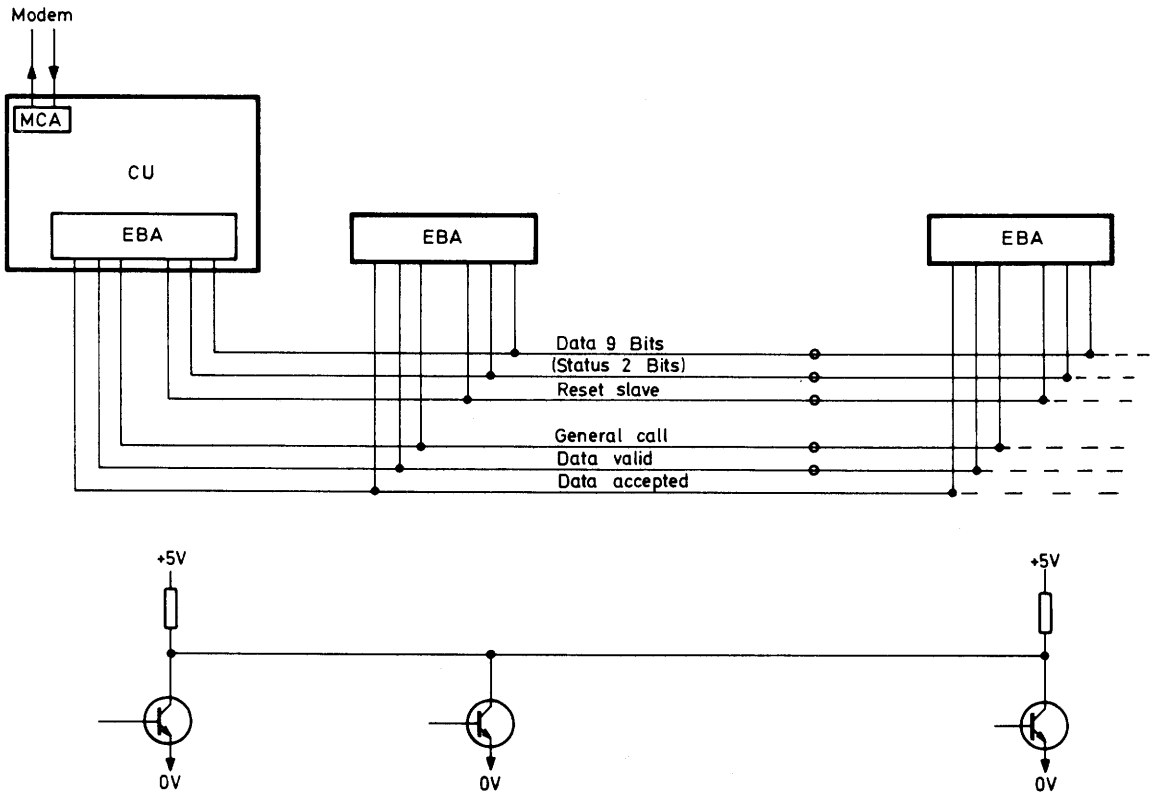


Fig. 1. External data bus (EDB)

DATA TRANSFER VIA EDB

The data transfer across a bus, when several unit can transmit, must follow well defined procedures to avoid confusion among the bus signals. These procedures are partly determined by the hardware design of the EBA, but are mainly determined by the programs.

In all procedures, the master is the only unit that can initiate data transfer. If a slave has a message to send to another slave, it so informs the master when the master makes a status poll. Then the master can give the slave permission to start transferring the message.

Control signals on the EDB

The transfers are controlled by the three control signals general call, data valid and data accepted. Each byte is transferred under control of the latter two. See Fig. 2. The master EBA (or a slave EBA that has got permission) sends out the signals general call and data valid (i.e. makes them low by making output transistors conducting). This forces all other EBA's into receive mode and they make the data accepted signal on the EDB low.

The transmitting EBA indicates that a valid byte is available on the EDB when the signal data valid is low. When the byte has been transferred to the MPU via the EBA in a receiving unit, the EBA at hand cuts off the output transistor feeding data accepted line. The data accepted signal on the EDB goes high when the last transistor is cut off. Thus, data accepted goes high when all receiving units have accepted the byte on the EDB.

When the transmitting unit senses the positive going transition in the data accepted signal, it raises the data valid signal. Then the data accepted signal is made low by those EBA's that are "interested" to continue receiving. In the other EBA's the output transistors will remain cut off, and these EBA's will not participate in the communication.

All bytes in the message are then transferred to the "interested" receivers. The transmitting EBA puts the byte onto the EDB, lowers the data valid signal and keeps it low until the last receiving EBA has accepted. When the last byte in the message is transferred the data valid signal remains high.

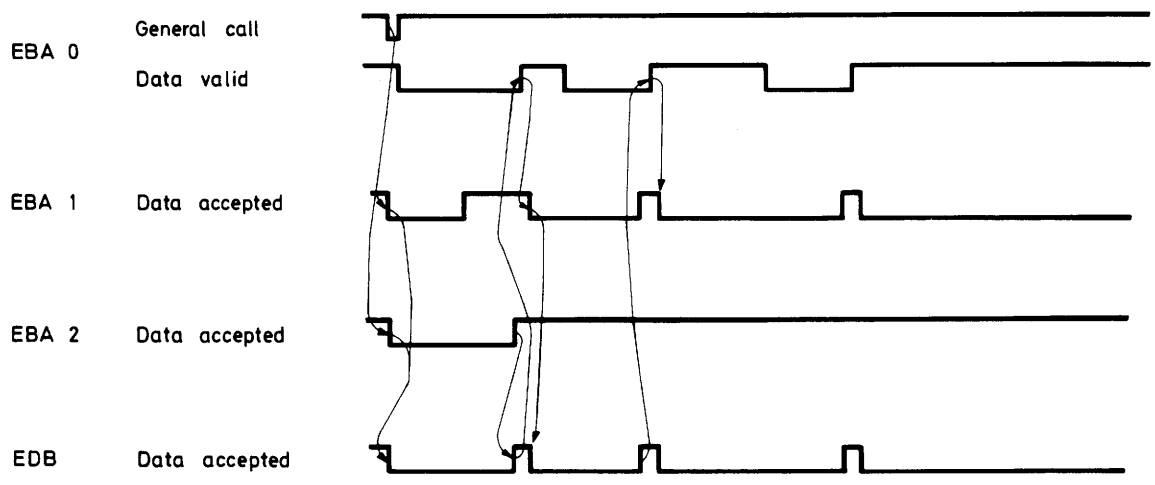


Fig. 2. Control signal timing

If one of the receiving units has an answer to send to the message, the EBA in this unit puts a byte onto the EDB and lowers the data valid signal. The data accepted signal goes high simultaneously (if there was only one receiving unit) but the former transmitter is not affected since it has not lowered the data valid signal. Instead, it lowers the data accepted signal because it is set in receive mode by the data valid signal from the former receiver. Note that the general call signal must not be lowered when answering. The programs control that two former receiver do not answer at the same time, and that a receiver does not answer before the transmitter has sent all bytes.

EBA operation during transfer

The block diagram in Fig. 3 shows the most important function blocks of the EBA. Signals to and from the control unit are drawn at the left side of the diagram and signals interfacing EDB at the right.

The address decoder decodes the instructions data from EBA (74 – INBS), data to EBA (64 – UTED) and status to EBA (65 – UTES) from the bits arriving on the I/O address bus. The bits of the status register can be set in any position via the I/O data bus. Bit 3 can be fed through the gate to send reset slave only in a master. Bit 2 and 1 are normally not used. Bit 0 is one-set in EBA's that are participating in communication. A general call on the EDB one-sets bit 0. The status register can be zero-set by general reset from MPU.

The timing diagram in Fig. 4 shows the time of events when a master is starting transfer (three bytes shown) to one of two connected slaves.

Transmitting general call

The master MPU first gives the instruction status to EBA to zero-set status:0. Then, the master MPU puts the first byte to be transmitted (normally an address byte) onto I/O data bus and gives the instruction data to EBA whereby the byte is loaded into the 9-bits data register.

The instruction also affects the data valid logic, wherewith the signal general call is sent on the EDB and the data in the register is gated out onto EDB.

Simultaneously, the transmit/receive flip-flop (in the transmit/receive control circuits) is set in transmit mode. Then, the transistor feeding data accepted signal will be kept cut off as long as transmit mode prevails.

When the output instruction data to EBA ends (after approx 1 μs) status:0 is one-set, as it shall be in a unit involved in communication. Then the data valid signal is lowered, and the general call signal disappears after a short delay.

Receiving general call

When the other EBA's connected to the EDB sense general call, they one-set their status:0. When they sense data valid, their transmit/receive FF is set in receive mode, whereupon they issue EBA interrupt to the MPU. Data valid also causes data accepted signal to go low.

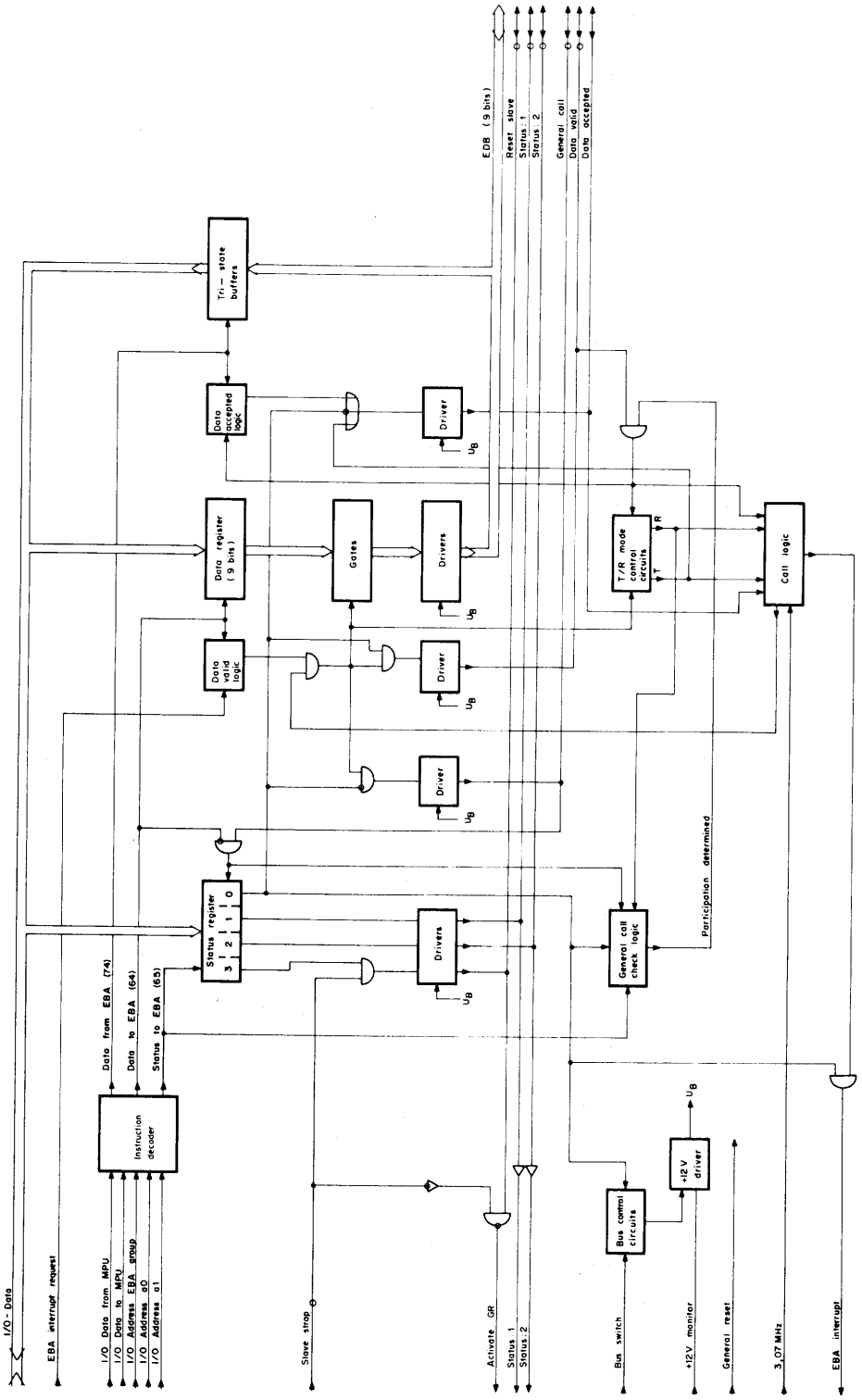


Fig. 3. Block diagram of the external bus adapter (EBA)

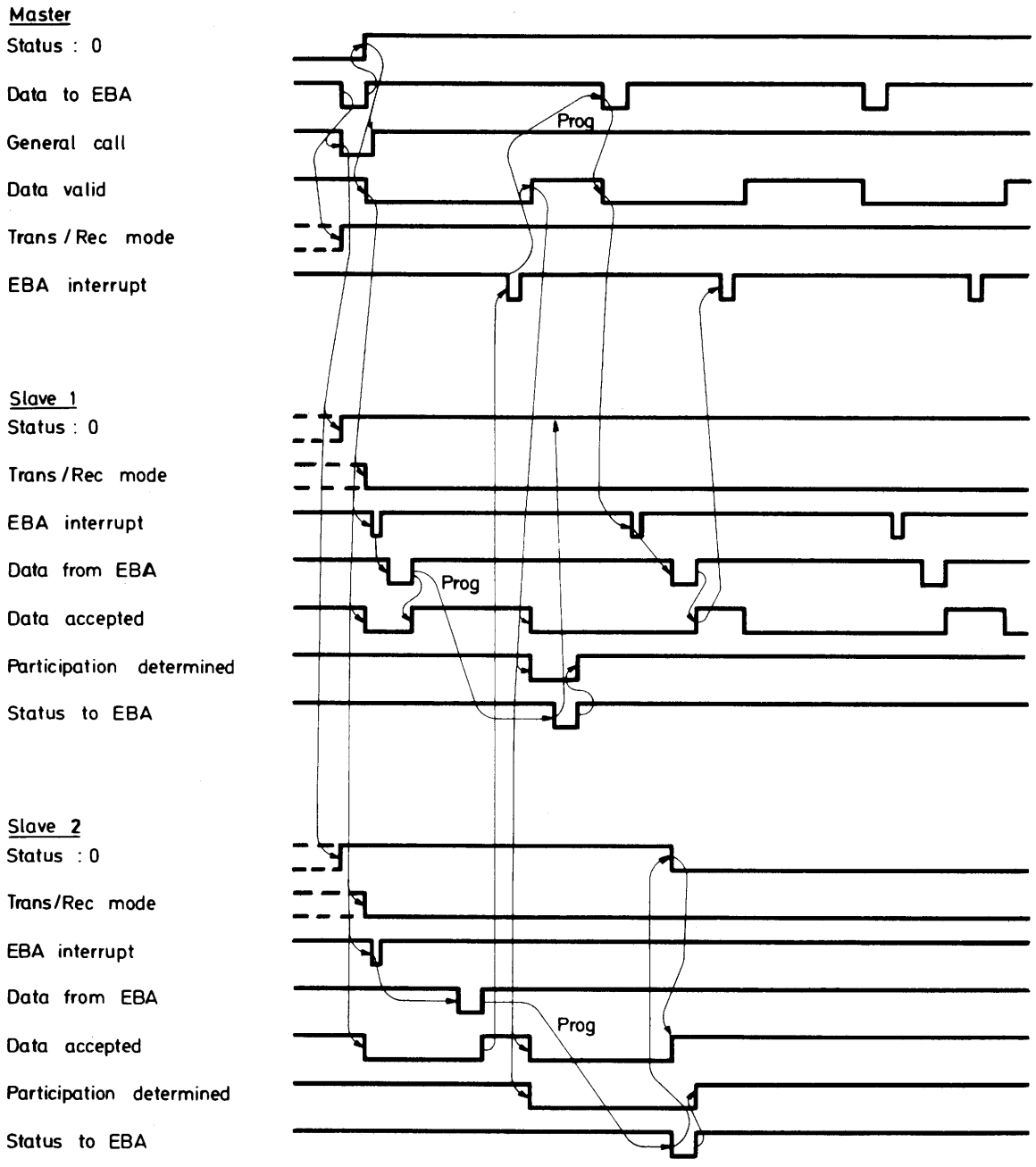


Fig. 4. EBA timing diagram

The program that is started in the MPU after the interrupt is received contains the input instruction data from EBA. The programs in different slaves start after different time delays depending on the MPU operation. The input instruction opens the tristate buffers so that the data bits on EDB appear on the I/O data bus. It also affects a flip-flop in the data accepted logic so that the output transistor feeding data accepted line is cut off. Thus, the data accepted line will go high when all slaves have transferred the byte to their MPU.

The call logic in the master EBA issues the signal EBA interrupt when the data accepted line goes

high. When the interrupt program in the MPU is initiated, the data valid signal goes high. When the slaves detect that the data valid signal goes high, they lower the data accepted signal and, via the T/R mode control circuits, lower the partipation determined signal from the general call check logic.

The program in the slave MPU's analyses the first byte received from the EDB. This byte should contain information (address and order) that allow the slaves to determine if they are to participate in the following communication or not. When the analysis is finished, the MPU gives the instruction status to EBA. The data on the I/O data bus then makes status:0 one-set (regardless it already is one-set) if the slave is to participate in communication or zero-set if it is not. If status:0 is zero-set the slave will be passive until it receives another general call.

The instruction status to EBA also makes the signal participation determined high via the general call check circuits. Simultaneously with the analysis in the slaves, the master MPU puts the next byte onto the EDB. Depending on how fast the interrupt signals are handled by the MPU three situations can occur:

a) If a slave is very fast it can give status to EBA while the data accepted line is still low (i.e. before the first interrupt to the master MPU). In such case, the participation signal never goes low. This situation is not shown in Fig. 4.

b) The slave EBA gives status to EBA after data accepted line went high but before master MPU has put the next byte on the EDB (slave 1 in Fig. 4). Situations a) and b) are normal.

c) The master MPU puts the byte on the EDB before a slave has analysed the first byte (slave 2 in Fig. 4). It is not allowed to issue another interrupt to the MPU before it has determined if it is interested to receive more bytes. (If it is not, the second interrupt would be ununderstandable to the program.) The data valid signal from the master is in this situation blocked in the gate until the participation determined signal goes high. As a consequence, the second byte on the EDB will not be accepted until then.

Transfer

If status:0 is zero-set, the transistor feeding data accepted line is cut off, the interrupt signal is blocked and the EBA at hand pays no intention to the signals on the bus (except general call). The master sends bytes that are received in all EBA that have status:0 one-set, as already described in section entitled "Control signals on the EDB".

The transfer is complete when the master MPU sends no more bytes. Nothing in the EBA is reset. The programs control the termination of the transfer e.g. by sending certain bytes.

Changing transfer direction

The transfer direction can be reversed under program control. When the slave has received the last byte, the program "knows" that it is free to send an answer. The MPU in the control unit that shall answer puts the first byte in the answer on the I/O data bus and gives the instruction data to the EDB. See Fig. 5. (The dashed lines indicate that the control unit at hand does not affect the control line.) Then, the EBA sends out the data valid signal (makes it low) and sets the T/R control circuits in transmit mode. Note that general call is not sent as status:0 is one-set.

However, there is no information in the program or in the MPU about the status of the other control units. Thus, it is possible that the last byte is not accepted by all other control units, which means that the data valid signal is still kept low by the master EBA. In such case, a flip-flop in the call logic is set so that the gate below the data valid logic is closed. This means that the data valid signal cannot be made low, from the slave EBA, until the master EBA has made it high.

When the master receives low data valid from the line, it is set in the receive mode, whereupon it issues EBA interrupt. The master MPU fetches the byte from the EDB and the communication then proceedes as usual with the slave as transmitter and the master as receiver.

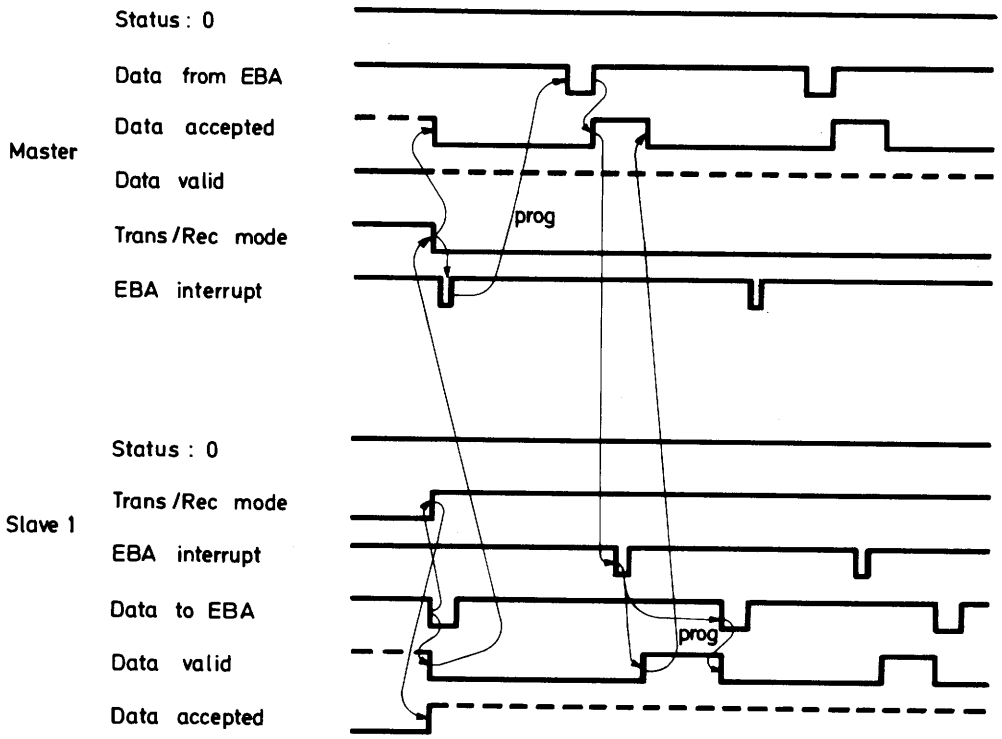


Fig. 5. Changing transfer direction

DETAILED DESCRIPTION

In the following detailed description of the external bus adapter circuitry, only a few basic situations are described. How these situations follow each other to perform a data transfer is described in the preceeding sections. The reader should be familiar with these sections before reading the detailed description. The descriptions of the situations are mainly based on timing diagrams that show the relevant events. Also other events can occur (i.e. other flip-flops are set), but they have no significance for the operation associated with the situation at hand.

Note that the only hardware difference of a master EBA and a slave EBA is due to that P68 is high in a master EBA and low in a slave EBA. This prevents other EBA than the one in the master to transmit the signal reset slave.

Transmitting general call

A connection for transfer is established by master MPU giving two output instruction via the I/O address bus: 65 (status to EBA) and 64 (data to EBA).

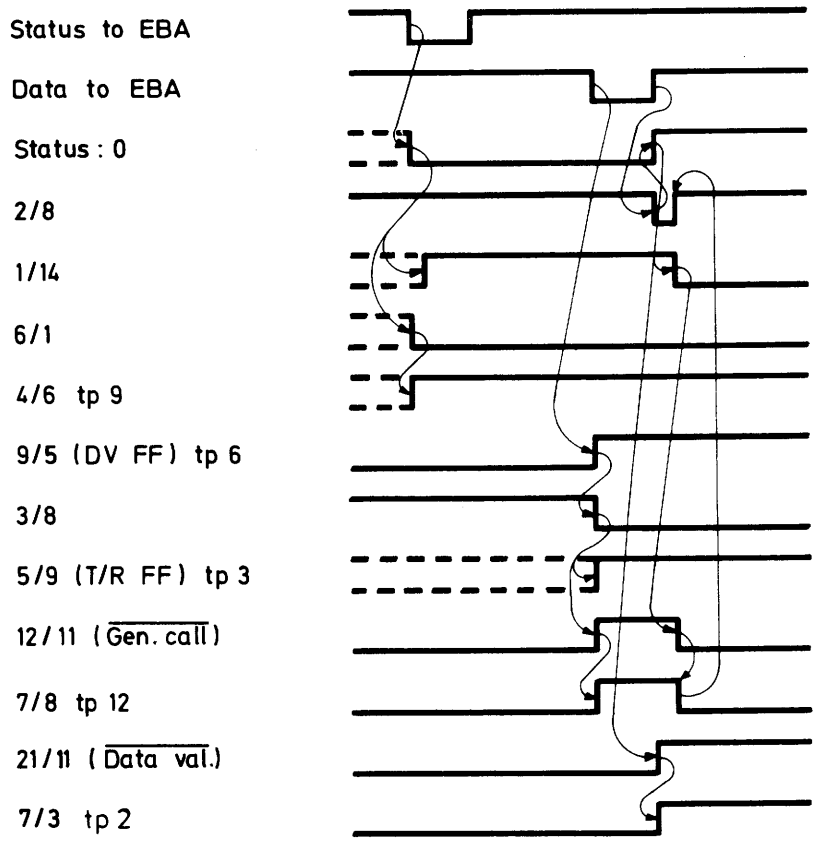


Fig. 6. Transmitting general call

In the first output instruction (65) data sent over the I/O bus must zero-set status:0, which is necessary to transmit general call. See Fig. 6. When gate 6/1 is low, the master does not contribute to make the signal data accepted low (Y11 cut off via 12/3).

The second output instruction causes the general call signal to be transmitted as can be seen in Fig. 6. Note that 6/1 is now kept low by the transmit/receive FF. Gate 2/8 gives a pulse when the signal data to EBA goes high again wherewith status:0 is one-set. This status bit must be one-set in all EBA that participate in data transfer. It must be one-set in response to the trailing edge of the signal data to EBA, otherwise the general call signal would be too short.

Receiving general call

When a general call is received from another EBA, the transmit/receive FF and status:0 can be in either position. The general call signal makes status:0 one-set (regardsless of former position), and the data valid signal sets the T/R FF in receive mode. See Fig. 7. When both these events have occurred, gate 6/1 will be high (if it was not before). Then the data accept FF will determine the data accepted signal.

The data valid signal also causes interrupt signal to the MPU that answers by sending the instruction data from EBA (74) via I/O address bus. Then data from the EDB is gated onto I/O data bus. The MPU then gives the output instruction status to EBA and the information sent to EBA one-sets (if the control unit at hand is to participate in the communication) or zero-sets status:0. In the latter situation, no more interrupt signals are issued and gate 6/1 causes transistor Y11 (DA) to be cut off (dashed lines in Fig. 7).

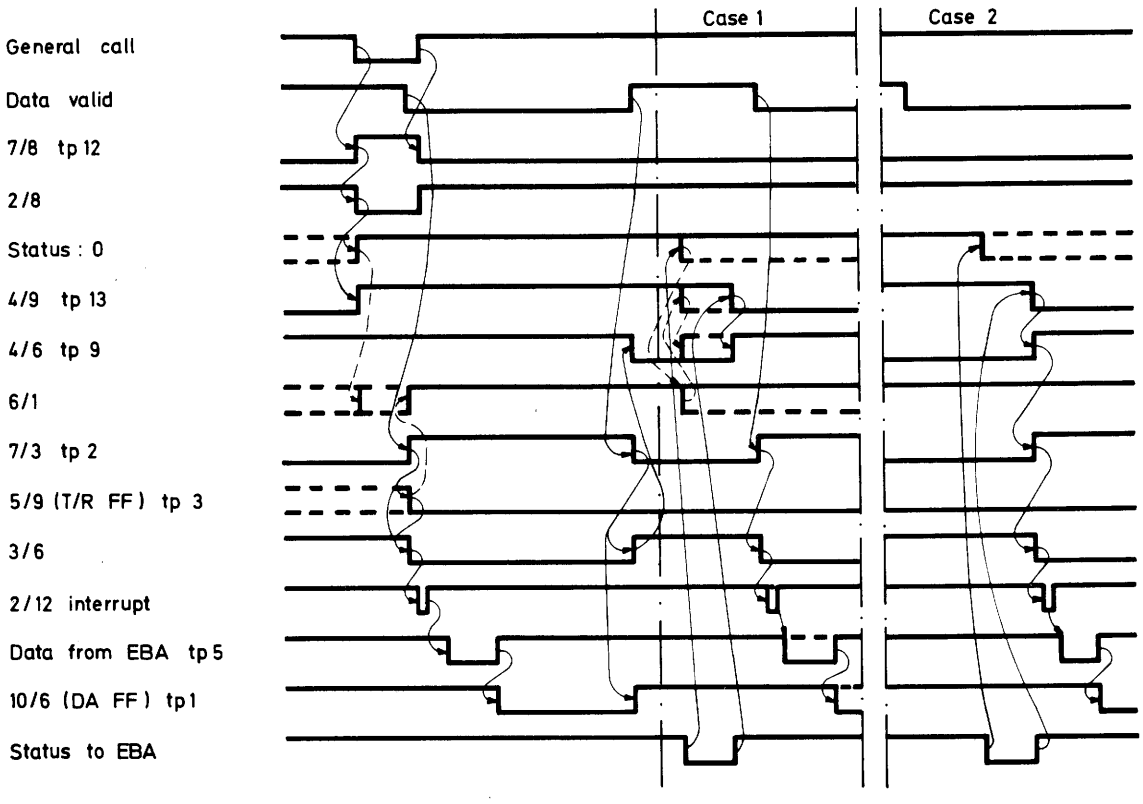


Fig. 7. Receiving general call

In the mean time, however, the transmitting unit has sensed that the data accepted signal went high and puts another byte onto EDB (and sets the data valid signal low). The first case in Fig. 7 shows what happens if status to EBA comes before this next high to low transition of the signal data valid, and case two shows the opposite situation. Of course, status to EBA can arrive while the data valid signal is still low, due to another slow EBA, that holds the data accepted signal on the bus low. However, this case does not make the situation different (from case 1 in Fig. 7) when the next data valid signal high to low transition occurs (as tp9 will be high after status to EBA).

From Fig. 7 it is evident that the EBA is unable to accept the byte after the one transmitted together with general call until the MPU has determined if the EBA shall participate in the transfer or not. Meanwhile, no communication can take place on the EDB.

Data transfer, transmitting unit

When a connection is established, the transmitting unit has status:0 one-set, and T/R FF in transmit mode. Then gate 6/1 is low (i.e. transistor Y11 is cut off).

The last receiving unit to issue the input instruction data from EBA, accepts the byte on the EDB by making the data accepted signal high. Then the events shown in Fig. 8 take place in the transmitting unit.

The data valid signal is made high when the MPU initiates the interrupt program. When the receiving units sense that the data valid signal is high, they lower the data accepted signal. After the MPU has fetched the next byte to transmit, it issues the output instruction data to EBA. Then the data valid signal goes low again, thereby informing the other units that the next byte is available on the EDB. The transfer can also be concluded without transmitting any new byte.

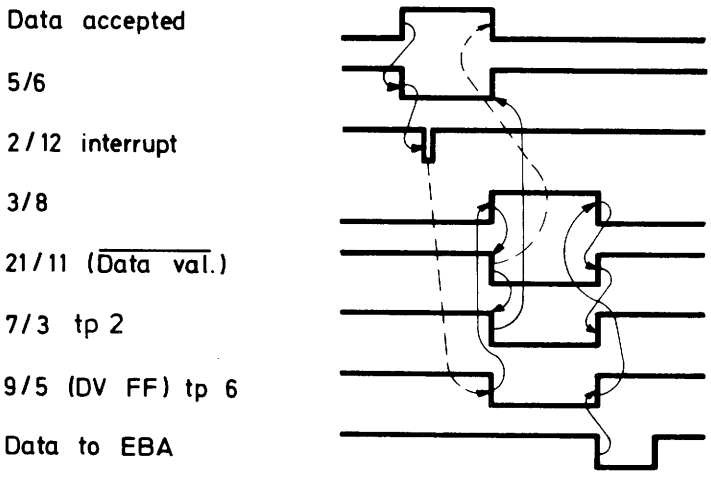


Fig. 8. Data transfer, transmitting unit

Data transfer, receiving unit

A receiving unit, that is established, has flip-flop 4/6 (tp9) high, the T/R FF in receive state and the DV FF (tp6) does not work. Fig. 9 shows how an interrupt signal is issued each time the data valid signal goes low.

When the MPU fetches the byte (by instruction data from EBA), the DA FF is set so tp1 is low. Thereby the EBA at hand informs that it has taken care of the byte. When the last of the participating receiving unit accepts the byte, the data accepted signal on the bus goes high. Then the transmitting unit raises the data valid signal, which causes resetting of the DA FF wherewith the data accepted signal goes low.

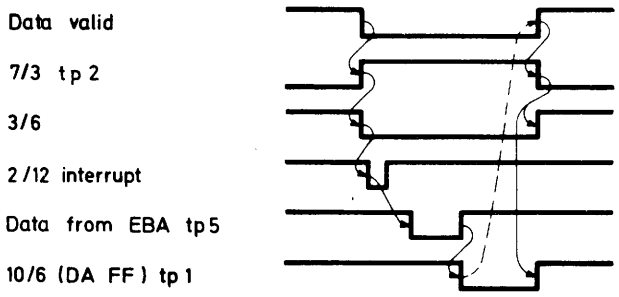


Fig. 9. Data transfer, receiving unit

Changeover from receiving to transmitting

If a receiving unit shall send an answer to the unit that has been transmitting, the MPU gives the output instruction data to EBA after it has taken care of the last received byte. Two cases can then occur: 1) The byte has been accepted by all receiving units and 2) The byte has not been accepted by all receiving units. (The first case is always prevailing if there is only one receiver.)

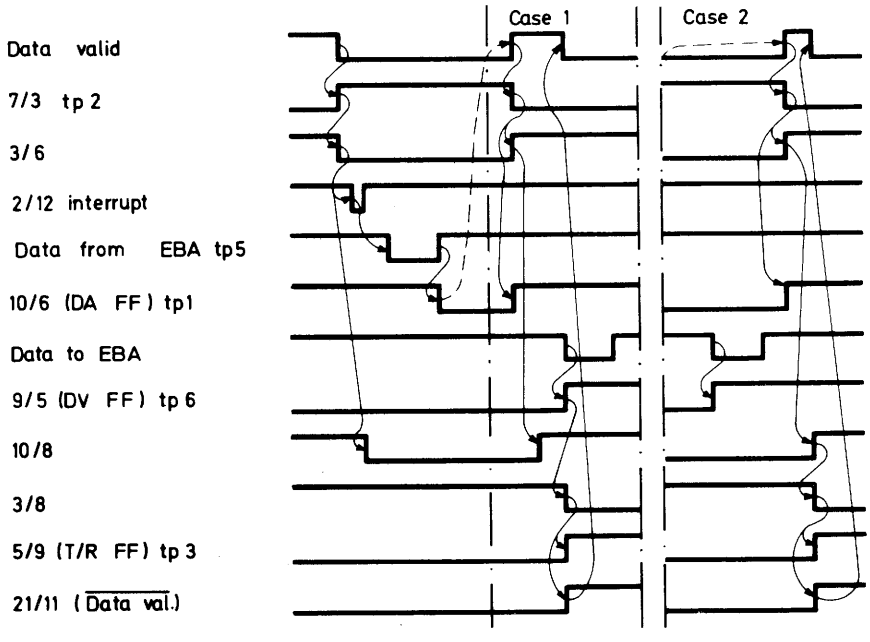


Fig. 10. Changeover from receiving to transmitting

The events that take place in these two cases are evident from Fig. 10. The upper left part of this timing diagram is exactly the same as the diagram in Fig. 9. Flip-flop 10/8 plays an essential role in connection with changeover. If all receiving units has accepted the last byte, the transmitting unit has raised the data valid signal. Then the output 10/8 is high when the MPU issues the output instruction. Thus gate 3/8 goes low setting the T/R FF in transmit mode and causing the unit at hand to lower the data valid signal. After that, the unit is in the transmitting mode.

If the former transmitting unit has not raised the data valid signal (because some other EBA still keeps the data accepted signal low), the output instruction only one-sets the DV FF. Flip-flop 10/8 is kept one-set (output 8 low) by the data valid signal (via gates 7/3, 3/6, 3/3 and register 1). The gate 3/8 then goes low after the data valid signal has been raised by the former transmitter, whereupon the changeover is completed as in case 1.

Changeover from transmitting to receiving

When a transmitting unit has transmitted its last byte, it must not receive any more output instructions data to EBA. The timing diagram in Fig. 11 shows the most essential events when the data accepted signal goes high. After the data valid signal is raised, the transmitting unit waits. When the data valid signal goes low (because a former receiving unit has become transmitting), the T/R flip-flop is set to the receive state via gate 7/3. Then the unit has changed over from being transmitting to receiving. (Note that DV FF is not set as the output instruction is absent.)

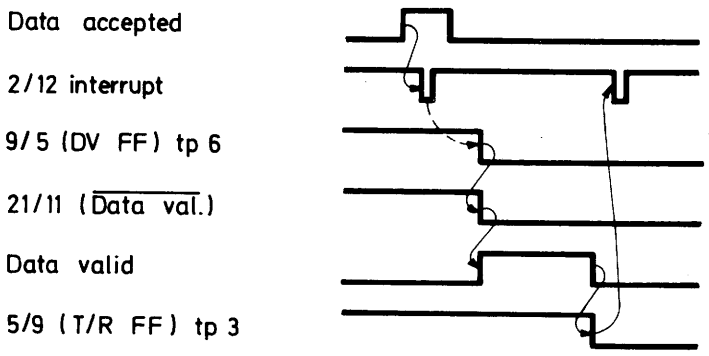


Fig. 11. Changeover from transmitting to receiving

Bus control circuits

The main part of the bus control circuits is shown in Fig. 12, along with a timing diagram showing the power on and off events. At power on, the bus switch may be in either position, and if it is in off position, it may be switched on at any time.

All line drivers are powered from +12 V via a transistor (U_B). This transistor will not conduct until +12 V monitor signal goes high. See Fig. 12. At this time, the EBA logic will have assumed its initiated conditions. The gates feeding the drivers have open collector (type 7409) to ensure that all lines in the external bus are unaffected until the +12 V monitor signal arrives. If the bus switch

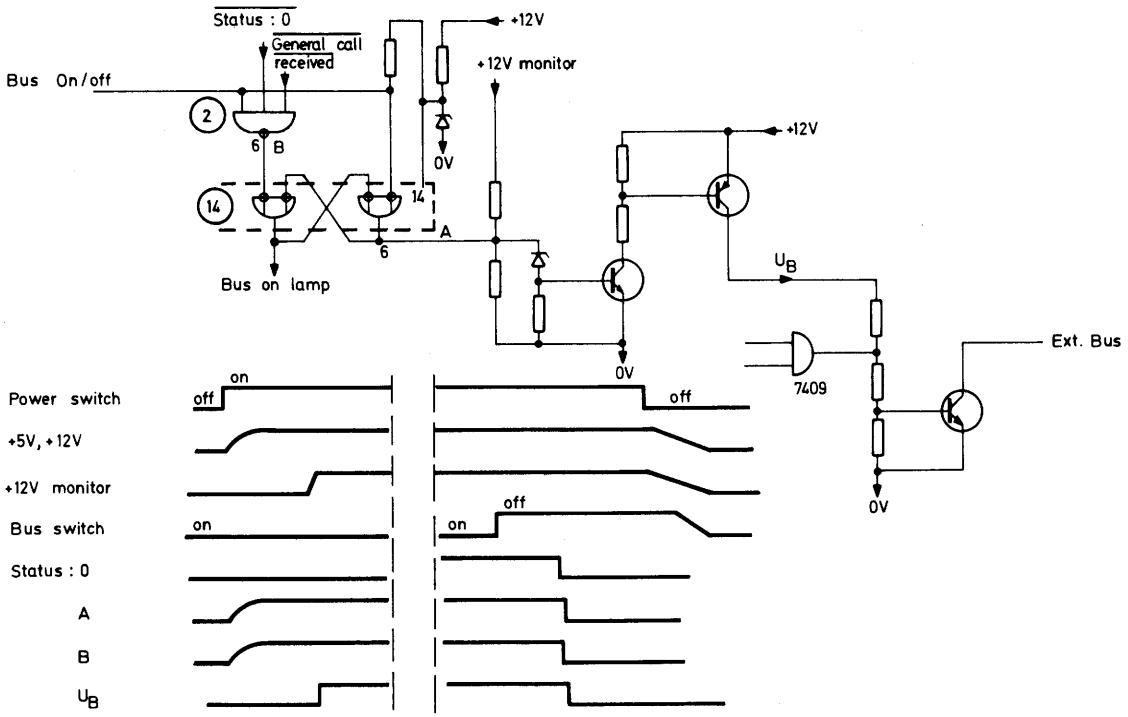


Fig. 12. Bus control

is in off position, the flip-flop will be forced to set 14/6 low, thereby cutting off the drive transistor.

The power switch must not be switched off while data is transferred on the external bus. First the bus switch should be set in position off. Gate 2/6 then goes low when the unit at hand is not participating in any data transfer. Then the flip-flop (14) is set so that the bus on lamp is extinguished. After that the power switch may be turned off.

Line Multiplexor

The Line Multiplexer 3534 is used if up to four control units / control modules should be connected to one modem. It comprises a small PC-board, having some integrated amplifiers and discrete components. The PC-board is assembled in a mounting angle which can be screwed to the rack of the cabinet.

The angle is provided with five modem connectors; P5 for the modem cable and P1–P4 for connection of frames 1–4. See circuit diagram. Note that frame 4 is the basic frame mounted at the lowest location in the cabinet.

The received data signal and the control signals from modem are amplified. The output amplifiers can feed two modem communication adapters (MCA). The transmitted data and request to send signals from the MCAs are amplified and OR-wired to the output amplifier.

The timing signals can be supplied either from modem (switch S1 in position EXT) or from the MCA in frame 4 (basic frame – switch S1 in position INT). See circuit diagram.

STANSAAB

Introduction

Mechanical Design

CRT unit

Serial Transfer Logic

Keyboard

Identification Device

Light Pen Adapter

Alfaskop 3500 Data Terminal System

Technical Description

Display Unit 3510

EE559–412
1975–01–10

Introduction

This part describes all components that the operator uses, the 3510 Display Unit, the 3540 Keyboard, the 3541 Identification Device and the 3542 Light Pen.

The basic component Display Unit 3510 is designed for cluster configurations. This unit is connected to a display adapter in Control Unit 3501/3502. The distance between a display unit and the centrally located control unit can range up to 600 metres.

The optional Keyboard 3540 is connected to the display unit via a one metre cable. The Identification device 3541 is mounted in the Keyboard. The Light Pen 3542 is connected to a light pen adapter which is inserted in the display unit.

The display unit comprises a CRT unit, a power supply (described in section CRT unit) and a serial transfer logic.

- The CRT unit is provided with an anti-glare screen that can be tilted up to 17^0 from a vertical plane. Scanning is carried using a modified TV raster method. The effective screen size is 258 x x 180 mm. Two brightness levels and also a non-visible level are provided.

- The serial transfer logic (STL) administers the data flow between the display unit and the control unit as well as within the display unit. It also controls the indicator lamps on the panel, and the alarm circuit that is triggered when the operator tries to make invalid inputs. The control functions of the STL are determined by commands received from the control logic.

The display unit is provided with the following controls

- Power on/off switch
- Brightness control for regulating display brightness
- Contrast control for regulating the difference between text fields having high and normal brightness
- Audible alarm volume control
- Click volume control (located under the keyboard) for regulating the volume of the clicking sound that acknowledges keyed entries

An indicator panel located at bottom of the front of the display unit informs the operator of the terminal's status. This panel contains the following indicator lamps

• Power on	Indicates that power is switched on to the terminal
• System Ready	Indicates readiness for traffic with the computer
• I/O error	Indicates an error in an external unit (a data printer for example)

In addition to the indicator panel lamps there are three indicator lamps on the keyboard, namely

• USM request	Mounted in the USM key. Indicates that the computer has an unsolicited message for the operator

- Insert mode

  Mounted in the INS MODE key. Indicates that this key has been depressed, i.e. that the display unit is operating in the insert mode

- ID

  Mounted at the top cover of the identification device. Used only when the identification device option is included and in such case it indicates that the ID card has been read correctly.

Mechanical Design

The picture tube is mounted in a frame which is attached to the cabinet of the display unit in such a way that the screen can be tilted up to 17^0 from a vertical plane. One small PC-board containing protective circuits is mounted on the picture tube. Two PC-boards (containing deflection circuit, video amplifier and stabilizing circuits) are mounted on the movable frame. This frame can easily be detached for servicing etc.

The mains transformer is mounted at the bottom of the cabinet and the voltage rectifier and stabilizing circuits are mounted on a PC-board at the left. At the right there is space for the light pen adapter PC-board. The serial transfer logic PC-board is mounted at the rear, accessible if the rear cover of the cabinet is removed.

The keyboard is a separate unit, connected to the display unit via a one metre cable, which is connected to the serial transfer logic inside the cabinet. Inside the keyboard it is one PC-board on which the dry reed switches as well as the logic circuits are mounted. At the rear of the keyboard there is space for mounting of the logic PC-board and the light emitting diode PC-board belonging to the identification device.

CRT Unit

Since the supply voltage sent to the different stages in the display unit/station affects the width and height of the picture that is displayed, and also affects all auxiliary voltages obtained via the line transformer, a voltage stabilizer has been included to handle any variations in the incoming voltage. The supply voltage is adjusted using trimmer potentiometer P200.

The CRT is a 110^0 thick-neck monitor tube of type M38–121W (Brimar or Philips equivalent).

The horizontal deflection circuit is directly controlled and, with the assistance of the line transformer, it simultaneously generates heater voltage, static focus voltage, screen grid voltage, control grid voltage, supply voltage to the video amplifier and high voltage. In order to avoid disturbances in the picture attributable to stray flux in the line transformer, it has been positioned as far away from the CRT neck as possible. The width of the image is adjusted by changing the supply voltage sent to the display unit/station (P200).

The vertical amplifier is also directly controlled and, in its output stage, it makes use of an integrated audio amplifier (IEK 300 TBA 800) provided with a protective circuit comprising BA 148 and BC 161 at the output. The vertical hold, vertical height and linearity can be adjusted by trimmer potentiometers P302, P300 and P301 respectively.

In order to obtain good focus throughout the entire image area, the focus voltage is varied during the horizontal sweep time – dynamic focusing. The focus voltage has a DC component (adjusted by P202) and a parabolic component derived from the horizontal sweep (adjusted by means of P201).

The brightness is regulated using an external potentiometer that regulates the control grid voltage supplied to the CRT. The brightness can be increased for a desired portion of the picture area by means of a control pulse sent to the video amplifier. The amount of increase in brightness is regulated by another external potentiometer.

The video amplifier comprises a cascade-coupled stage (Q308 and Q309) which, in principle, functions as a switch connected to the CRT cathode. The black level corresponds to the high level (+70 V), and when the input signal reaches its end limit, the cathode is switched to the bottom voltage of the video stage. The bottom voltage is fixed (approximately +20 V) for normal brightness, but can be made variable (lower) when the control pulse for increased brightness is made higher by means of Q306 and Q307.

The circuitry is built up on three printed circuit boards. One printed circuit board contains protection circuits for CRT flashover as well as the tube holder. This printed circuit board is mounted on the CRT. The two other printed circuit boards are mounted on either side of the CRT neck. The voltage stabilizer and horizontal deflection circuitry are built up on one of these printed circuit boards; the vertical deflection and video amplifier on the other. A cable harness joins the three printed circuit boards and the connector used for input signals and the supply voltage. The cable harness is connected to the printed circuit boards via cable/PCB connectors. The deflection coils are connected to the printed circuit boards by means of pin/socket connectors. The printed circuit boards can be removed by disconnecting these cable/PCB connectors and pin/socket connectors and

removing three screws.

POWER SUPPLY

The power supply delivers three galvanically isolated voltages.

a) Unregulated supply voltage to display unit/station
b) 5 V stabilized voltage
c) 12 V stabilized voltage

Brief description of stabilizer circuits

In each of the stabilizer circuits, the collector on the series transistor is turned towards the load in order to obtain maximum efficiency. The only parameter that determines the lower limit of the voltage drop across the stabilizer is the bottom voltage of the series transistor.

Both units are protected against short circuits and are provided with a negative-slape current-limitation transition. Since current is sensed by measuring the base current in the series transistor in order to save power, current limitation depends upon the transistor's small signal short circuit forward current transfer ratio. For this reason, provision has been made for adjustment. In the 12 V stabilizer, the current limitation depends somewhat on the mains voltage since the series transistor is of the PNP type.

The current limitation does not normally require adjustment.

Serial Transfer Logic

Contents

GENERAL 1

BRIEF OUTLINE 1

DETAILED DESCRIPTION 5

- Sync separation 5
- Command byte 6
- Load data, read 7
- STL priorities 7
- Data byte 8
- Video and intensity control 8

FIGURES

1. The first 24 sweeps of the frame 2
2. Data bit timing 2
3. Sync, data and video transfer 2
4. STL block diagram 3
5. Principle of two way transfer 5
6. Sync Separator, time diagram 6
7. STL main timing 10

GENERAL

(Also consult DA, TIM and MUX sections.)

The information between controller and display is transferred in series via one coaxial cable. The controller feeds video, sync pulses (V and H), brightness control pulses and certain commands to the display. The display transfers keyboard-, identification- and light-pen data as well as certain other data to the controller.

Since only one conductor is used, it is obvious that it must be shared in some form of time multiplex. The exchange of information is made by the Serial transfer logic, STL, and the display adapter, DA.

- STL contains one PC board, housed in the rear of the display. It interfaces the DA via the coaxial cable and the CRT, the ID logic, the light-pen logic and the keyboard via multiwire cables.

- DA is assembled on a PC board in the controller. It interfaces STL, MPU via MUX and TIM.

BRIEF OUTLINE

To make 312 sweeps takes 20 ms and each sweep 64 μs. The main part of the screen time is occupied by feeding video and thus generating the text i.e. 288 sweeps.

Each frame needs a vertical hold pulse of 700 μs, which is generated in STL. The timing of this is controlled by the program in TIM which feeds a 58 μs pulse every 20 ms via DA to STL. This pulse defines the start of a new frame. STL converts the pulse and feeds V-hold of 700 μs to the CRT part.

Each sweep needs a horizontal hold pulse of 28 μs. This is also generated in STL, which is triggered by a 2 μs pulse, generated in TIM and fed via DA.

The first 24 sweeps of a frame are not used for text generation. During these 24 sweeps, V-hold (58 μs) and three bytes are exchanged on the coaxial cable. Each sweep *one* bit is transferred and figure 1 shows the arrangement of the data exchange.

Note that during the transfer of byte 1 and 2, the signal direction is from the display to the controller, except during a short interval when the horizontal hold pulse is to be fed to the display. This interval is defined by the TIM program and contains the time between position 85 of the previous sweep and position 15 of the sweep at hand. Also see section TIM.

DA has an input/output register for shifting byte 0, 1 and 2. The bits of byte 1 and 2 are clocked into this shift register by position 53 from TIM. This position thus appear approximately in the middle of the transferred bit.

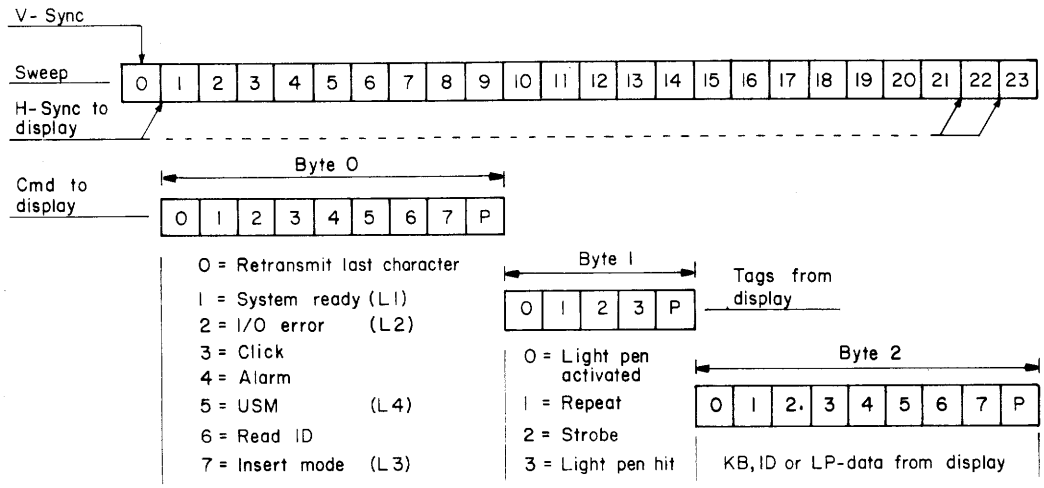


Figure 1. The first 24 sweeps of the frame

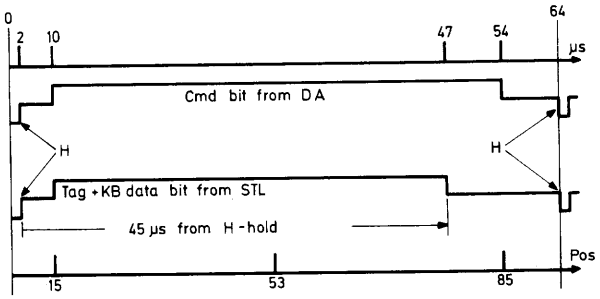


Figure 2. Data bit timing

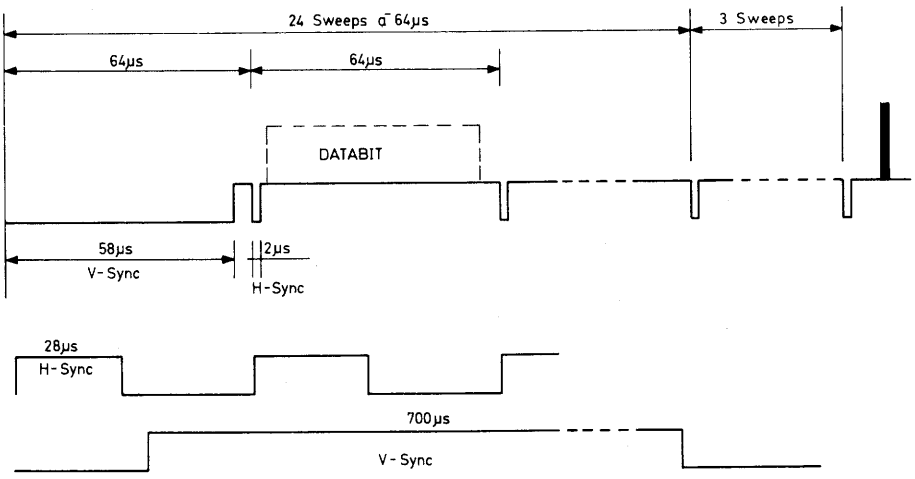


Figure 3. Sync, data and video transfer

Consider the block diagram, figure 4 and DA block diagram.
The MPU addresses DA via MUX and enters valid command data into the command register via I/O data bus. The command byte (0) gets correct parity and is set in parallel into DA I/O register.

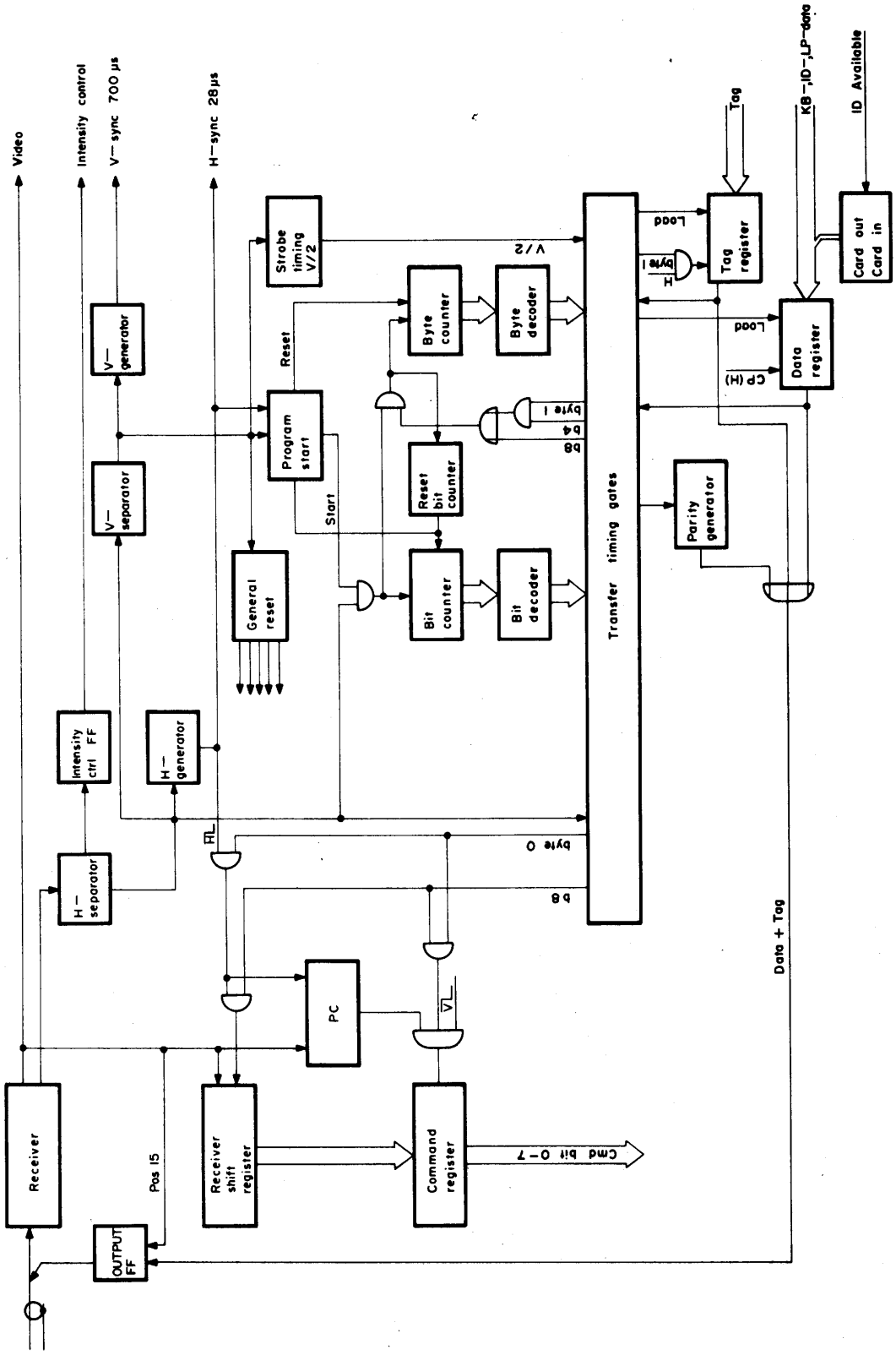


Figure 4. STL block diagram

At the beginning of each frame the TIM feeds a vertical hold pulse to the coaxial driver in DA. This defines sweep 0. The low level V-pulse is received in STL and fed to a sync separator, which decides that a V-pulse has been received. The separator controls a sync generator, which at this state generates both a vertical hold pulse of 700 μs and a horizontal hold pulse of 28 μs for the CRT-part.

At the detection of the V-pulse, the bit counter and the byte counter in STL are released. The bit counter is then clocked each H-pulse. At the beginning of each sweep the 2 μs H-pulse is fed to STL where it is detected and converted to a 28 μs pulse in the sync generator.

During nine sweeps after sweep 0 (V-sweep), the command byte (0) is fed to STL. The byte, which was set into the I/O register in DA, is shifted out by the H-pulse and thus H-hold and one bit is fed to STL each sweep. The bits are shifted into STL receiver shift register by the trailing edge of the H-pulse. The leading edge clocks the bit counter. Each bit is sensed by the PC–FF and if the parity bit agrees with the FF, the command is set into the command register.

The first ten sweeps has thus passed and the following 14 are used to transfer tags and KB-data from STL to DA.
At the beginning of sweep ten the H-pulse is fed to STL as usual. The bit counter in STL is stepped to ZERO and the byte counter to ONE, indicating that the tag byte is to be shifted out. At position 15 from TIM, DA blocks the transmission circuits in order to receive the first bit of the tag byte. Position 15 is also fed as a positive pulse to STL, where it is used as Cp for the output FF. The output FF is cleared about 45 μs after the appearance of the H-pulse. This is made from a timing circuit in the sync separator and defines the end of the data bit from STL. The bit at hand thus appear about 37 μs and in the middle of this time, position 53 from TIM is used as Cp for the I/O register in DA. Position 53 thus clocks the bit into the register.

At position 85 TIM opens DA transmission circuits again to prepare for the next H-pulse. See figure 2 and 3. This lap is repeated five times (sweeps) and then DA has received the tag byte and shifted it into the I/O register. It is then transferred to the status register, which is connected to MUX status register via DA bus.

Should the tag byte have a parity error, this is detected and the following data byte is cleared from the data register. This means that MPU fetches NUL-character with wrong parity.

The following nine sweeps, STL byte counter indicates byte 2, the KB-data is transferred and at the end of sweep 23 the byte is set from I/O register into the data register. DA interrupt then informs MUX that a character must be fetched.

Immediately after, H-pulses and video is fed to the display from DA via the coaxial cable.

Figure 5 shows the use of a single coaxial cable. Sync pulses and brightness control pulses are transferred with negative polarity and data, video and position 15 with positive polarity. The signal block video amp mainly contains (byte 1 + 2); (pos 15 to pos 85).

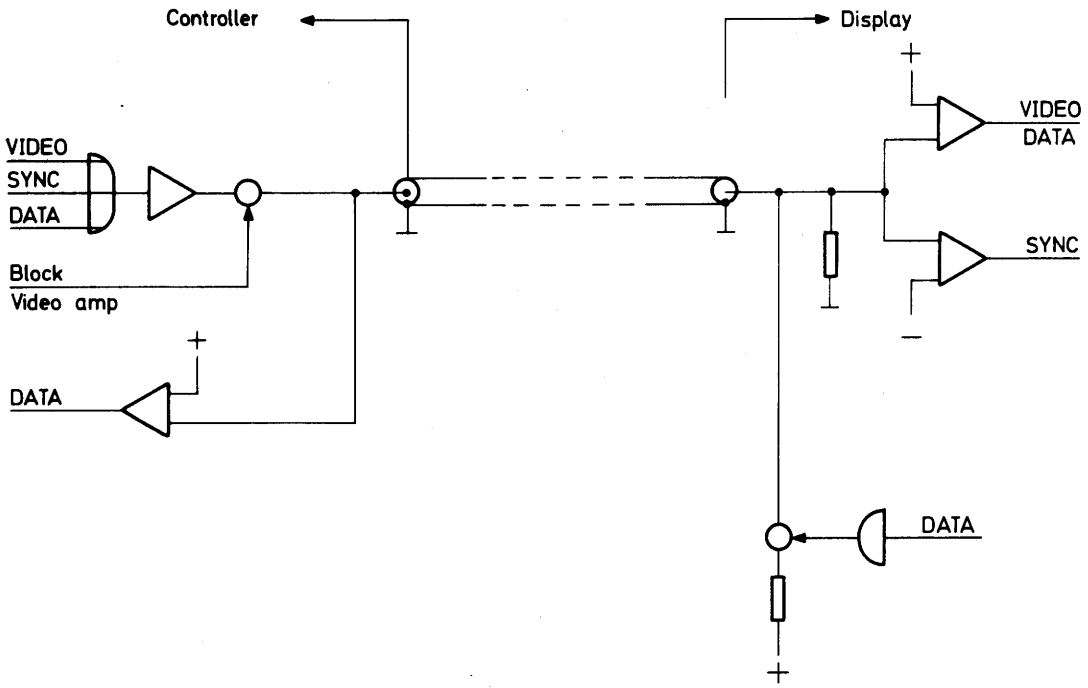


Figure 5. Principle of two way transfer

DETAILED DESCRIPTION

Sync separation

The low level (−12 V) V-hold pulse of 58 μs enters IC58 and triggers the monostables IC1, IC10, the V-hold Separator, is thereby set during 45 μs and since the reset condition at 9/10 has high level this FF is set at the end of the 45 μs. The ONE-side 9/5 triggers the V-sync generator IC11, which provides the 700 μs V-sync.

The monostable IC1 the H-hold Separator provides a 58−1.5 μs pulse, which in turn sets the H-sync generator IC2. Monostable 2 provides the 28 μs H-sync and 9/12 is reset at the trailing edge of the V-hold pulse from 58/4.

At the beginning of the next sweep, sweep 1, a new H-hold pulse appears. The monostables IC1 and 10 are set again but when 9 is reset after 2 μs, 9/3 has low level and the V-sync Generator cannot be triggered and no more V-sync is generated until the next V-hold is fed from the Controller.

The monostable 1 is able to set FF 9/12 by the trailing edge of its 0.5 μs pulse, since the H-hold pulse still has high level at 9/13. The H-sync Generator is therefore triggered and a new 28 μs pulse is applied to the TV-part.

Note that the V-hold Separator IC10 is set every sweep. This is used to provide the "end of bit" after 45 μs by resetting the Output FF IC26/11. Also note that 2/1 clocks the receiver shift register at the trailing edge of the 28 μs H-pulse during byte 0.

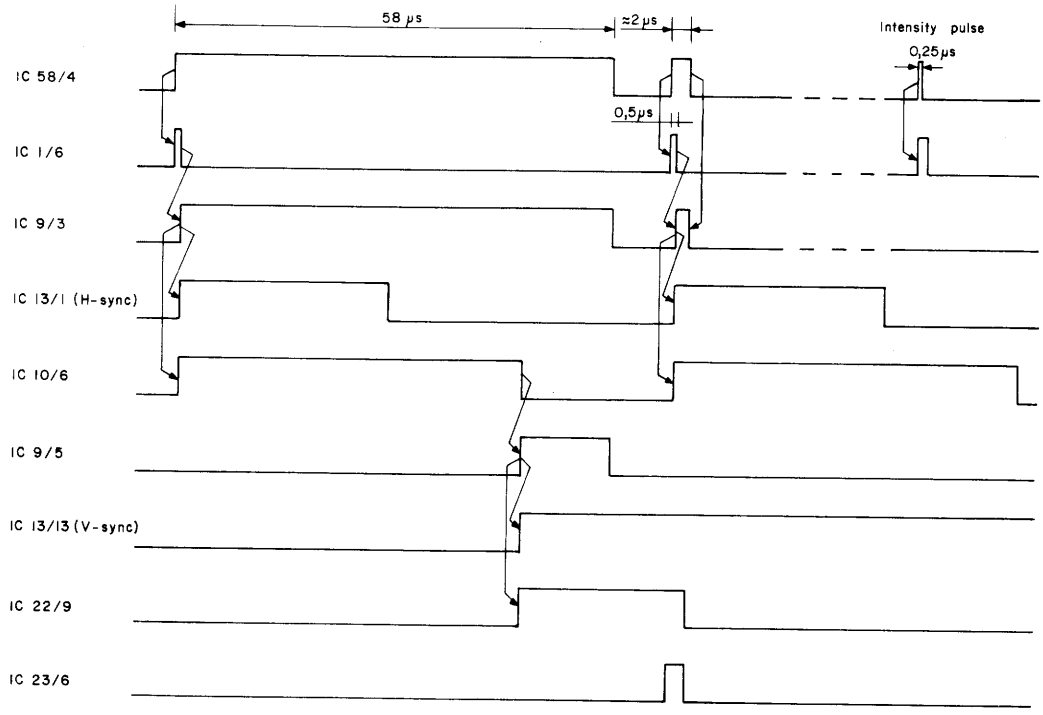


Figure 6. Sync Separator, time diagram

Command byte

The Program Start FF:s, IC22/11 and 35/12 are set by the V-hold pulse via IC9/3 and 5 and thereby the Bit Counter, IC17, and the Byte Counter IC3, are set to ZERO. The Bit Counter is then clocked by the H-hold pulses. Bits 0, 1, 2, 3, 4 and 8 are decoded in IC18 and Byte Counter is stepped by bit 4 during byte 1 via 20/3 and 6 and by bit 8 during byte 0 and 1.

During Byte 0, the logic receives the command byte from the Controller and the bits are shifted into IC41 by the trailing edge of the H-hold pulses, which are gated to the Cp input by byte 0 in 43/11. The parity of the Command byte is checked in 59/11 and if the parity is correct, the Commands are set into the Command Register, IC42 and 28 by the gated signal from 44/12.

During Tag byte, sweep 10–14, byte 1, is transmitted to the controller. The byte is set into the tag register, IC39, every other frame (V/2 from 33/8) at the end of byte 0 (byte 0 · step byte counter). However, read ID and retransmit strobe is set every frame. The byte contains

- LP activated
- Repeat
- Strobe
- LP hit

LP activated is a ONE when the light-pen is activated and LP hit is a ONE when the light-pen reaches coincidence at ascertain sensitive field, see light-pen description. These tags blocks the setting of strobe and repeat.

Repeat is a ONE when a move cursor key or a tab key is depressed more than 0.5 seconds on the keyboard, thereby causing automatic repeat.

The strobe bit is stored in 33/3 and 59/3. 33/3 is set at V-sync if ID-strobe or KB-strobe is applied (52/11) and 59/3 is set by retransmitt, ID-strobe or KB-strobe.

Note: The signal step KB is only used when the KB is provided with a character buffer. ID-strobe is generated either from a read ID command or when the card in/out logic IC25, 22/3 decides to transfer "card status".

The tag byte is shifted by H · byte 1 and the tag register output is connected to the output FF 29/11. The output FF is clocked by pos 15 · start FF35/3 · byte 0 and is cleared by the trailing edge of the V separator monostable 10/6. This means that the bit from the output FF is fed to the coaxial between pos 15 and the trailing edge of the detector pulse from 10/6, which appears approximately 45 μs from H, see figure 2. The last tag bit is fed from the parity generator FF26/3.

Load data, read

The output FF is connected to the latch FF:s IC29 and 30. IC29 detects when repeat or strobe is transferred from the output FF. Thereby 46/12 is opened and a load pulse is fed to the data register, IC48, via 46/6 as well as a clear pulse for the strobe FF:s via 45/8. Keyboard data is fetched by the signal read KB, which is generated in 32/6 and 56/6. IC29 is cleared by the succeeding Bit 0 (of byte 2).
The other latch FF, IC30 detects when the tag LP hit is transferred from the output FF. Thereby the signal read LP is generated as well as a load pulse in 56/8, and 46/6. Read LP and load data register thus enters the content of the LP counter into STL data register. IC30 is cleared by the succeeding bit 8 (of byte 2).

Data concerning ID is set when the command read ID is valid or when the card in/out logic feeds ID-strobe.
Read ID is gated by b_5 · byte 1 in 56/3 which feeds a read pulse to the ID-logic and, via 20/8 and 46/6, a load pulse for the STL data register. Note that no strobe is used during ID-byte transfer.

The card in/out logic detects when the low level active signal ID available changes state. ID available is used as data to 22/2, which is clocked once each frame. The card out FF, 25/3, and the card in FF 25/11, are cleared by general reset or "click" from MPU, and are clocked by ID available. Their outputs are gated in 4/3 and 11 by 22/3. Both 4/3 and 4/11 generates an ID-strobe, which sets the strobe FF:s 33/2 and 59/3 via 5/1, 52/11 and 52/6. At card in 4/11 provides high level to 49/13 and when the ID-strobe has been transmitted in the tag byte, a ONE is set into the data register as bit 5. This is interpreted by MPU as card in.

When the ID-card is pulled out, the card out FF, 25/3, is set and a pulse is fed from 4/3 as ID-strobe. When the strobe is transferred, 49/4, 2 and 6 feeds ONES as bits 9, 5 and 1 to the data register. This is interpreted as card out by MPU. The same function is also achieved (strappable) from 16/9 at general reset.

STL priorities

1) Read ID
   - Byte 1 is NUL with correct parity
   - Blocks Read KB and load KB data
   - Strobe and repeat are not transferred and thus load KB data is blocked
   - Blocks Read LP and load LP data
   - Blocks Read card in/out character
   - Blocks step KB
   - Blocks setting of strobe FF at retransmit
   - Transmission every 20 ms

2) Card in/out characters
   - Blocks LP act, LP hit and repeat
   - Blocks Read KB but not load data
   - Blocks Read LP and load LP data, LP hit is not transferred and thus read and load LP are blocked
   - Blocks step KB
   - Transmission every 40 ms

3) Light-pen
   - LP act and LP hit blocks repeat
   - Blocks Read KB and load KB data
   - Blocks step KB; strobe FF output and condition for step KB are blocked
   - Transmission every 40 ms

4) Retransmit
   - Blocks step KB
   - Transmission every 20 ms

5) New KB data
   - Transmission every 40 ms

Data byte

Byte 2 is set either from KB, ID, LP or card in/out logic. The byte is transferred in the same manner as the tag byte.

Video and intensity control

The video is fed via the coaxial and 58/9 to the CRT part. During the data transfer it is blocked in 24/8 by Cmd + Tags + data.

The video can be displayed with normal or increased intensity. This is controlled from the display adapter in the controller, which has certain field control characters stored in the display memory. These characters are decoded in DA and two of them, normal and high intensity, are fed as negative pulses via the coaxial to STL. The pulses must not be mixed up with the H-pulses and therefore they are only 0.25 μs. The H-pulse is 2 μs. The intensity pulses clocks the intensity control FF60/11, which is coupled to "divide by two". The flip-flop is cleared by H from 9/2 every sweep. The result is that 60/9 is cleared by H = normal, every sweep. When another negative pulse appears, 60/9 is set = high, until another intensity pulse is applied, setting 60/9 to ZERO = normal, or the succeeding H clears 60/9 = normal.

It is possible to strap DA for line wrap around of the high/normal function. This means that a "high" character can be written on one line and the succeeding "normal" character on a line further down. The intensity control FF is in this case still cleared by H, but it is set again by a negative pulse from DA which is generated during pos 15 from TIM. This pulse is of course only applied on the sweep between "high" and "normal". Note that pos 15, which is used as Cp for the output FF during byte 1 and 2 is a positive pulse on the coaxial and the set pulse during line wrap around is negative, similar to the sync pulses V and H.

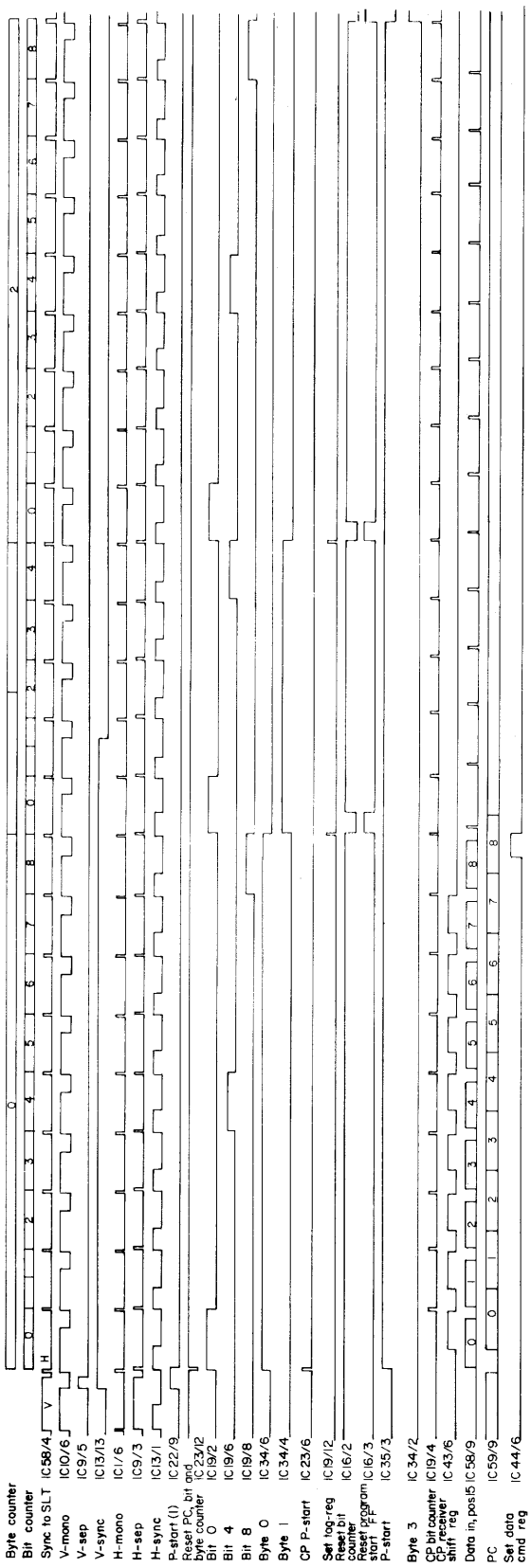


Figure 7. STL main timing

Keyboard

Contents

GENERAL	1
BRIEF OUTLINE	1
DETAILED DESCRIPTION	3

FIGURES

1.	Example of Swedish Keyboard code table	1
2.	Block diagram of Keyboard	2

GENERAL

The Keyboard consists of the punched key frame, which contains all key switches and the logic. The switches are soldered directly onto the PC-board. The switches are of dry reed type with interchangeable keytops.

The logic provides a code, corresponding to a depressed key and figure 1 shows the codes provided in hexadecimal. Certain code conversions are made by MPU.

KB

Bit No	8 7 6 5	0 0 0 0	0 0 0 1	0 0 1 0	0 0 1 1	0 1 0 0	0 1 0 1	0 1 1 0	0 1 1 1	1 0 0 0	1 0 0 1	1 0 1 0	1 0 1 1	1 1 0 0	1 1 0 1	1 1 1 0	1 1 1 1
4 3 2 1	C R	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0 0 0 0	0			65 93 SP	22 U 94 0	24 S É	36 S P	24 U é	36 U p	22 S =		72 U ⇧	72 S ⇧	71 U ⇖	71 S ⇖	4 U P1	4 S P6
0 0 0 1	1			13 S !	13 U 90 1	41 S A	27 S Q	41 U a	27 U q			73 U ⇩	73 S ⇩	70 U ⇘	70 S ⇘	5 U P2	5 S P7
0 0 1 0	2			14 S "	14 U 91 2	58 S B	30 S R	58 U b	30 U r			74 U ⇦	74 S ⇦	10 U KOPIA	10 S KOPIA	6 U P3	6 S P8
0 0 1 1	3				15 U 92 3	56 S C	42 S S	56 U c	42 U s	15 S §		75 U ⇨	75 S ⇨	1 U ÅTER	1 S ÅTER	7 U P4	7 S P9
0 1 0 0	4				16 U 86 4	43 S D	31 S T	43 U d	31 U t	16 S ¤		-26 85 U ⇨	26 85 S ⇨	66 U INS TKN	66 S INS TKN	8 U P5	8 S P10
0 1 0 1	5			17 S %	17 U 87 5	29 S E	33 S U	29 U e	33 U u			81 U ⇦	81 S ⇦	67 U BORT TKN	67 S BORT TKN	9 U SAND	9 S SAND
0 1 1 0	6			18 S &	18 U 88 6	44 S F	57 S V	44 U f	57 U v					68 U INS RAD	68 S INS RAD	3 U MEDD VAHT	3 S MEDD VAHT
0 1 1 1	7				19 U 82 7	45 S G	28 S W	45 U g	28 U w	18 S /				69 U BORT RAD	69 S BORT RAD	12 U	12 S
1 0 0 0	8			20 S (	20 U 83 8	46 S H	55 S X	46 U h	55 U x					76 U ↑	76 S ↑	11 U	11 S
1 0 0 1	9			21 S)	21 U 84 9	34 S I	32 S Y	34 U i	32 U y					80 U ⇦	80 S ⇦	2 U	2 S
1 0 1 0	A			52 5 89 *		47 S J	54 S Z	47 U j	54 U z	52 U '						77 U	77 S
1 0 1 1	B			23 U +	81 S ;	48 S K	51 S Å	48 U k	51 U å	23 S ?							
1 1 0 0	C			61 U ,	24 U <	49 S L	50 S Ö	49 U l	50 U ö	24 S >						79 U	79 S
1 1 0 1	D			63 U -		60 S M	37 S Ä	60 U m	37 U ä	63 S —							
1 1 1 0	E		79 ST	62 U 95 .		59 S N	38 S Ü	59 U n	38 U ü	62 S :							
1 1 1 1	F			39 U		35 S O		35 U o		39 S							

Figure 1. Example of Swedish Keyboard code table

BRIEF OUTLINE

See logic diagram. Figure 2.

The Keyboard is equipped with 92 code generating keys, two shift keys and two illuminated keys.

The encoded keys provide an eight bit code plus odd parity, which is fed in parallel together with a strobe pulse. The strobe pulse to STL is about 5 μs long. The code is maintained in the Output Register until the next key is depressed. Repeat signal has individual line output.

The logic holds

- an Oscillator
- a four bit Row Counter
- a four bit Column Counter
- a Decoder connected to the Column Counter
- a Selector controlled by the Row Counter
- an Output Register with a Parity Generator
- an Automatic Repetition Circuit
- a Bounce Block Circuit

The basic principle is to stop the two counters by pressing a key. The binary value of the counters, when stopped, then represents the code for the specific key.

The keyboard has automatic repetition facility for the move-cursor-keys and tab-keys. A special feature provides the two key rollover, required to prevent errors, caused when typing fast which occur on common letter diagrams such as TH and HE. E.g. a second key is often depressed before the preceding key is completely released. The rollover feature gives the correct output from both keys in sequence, without slowing the typist or causing error in entry.

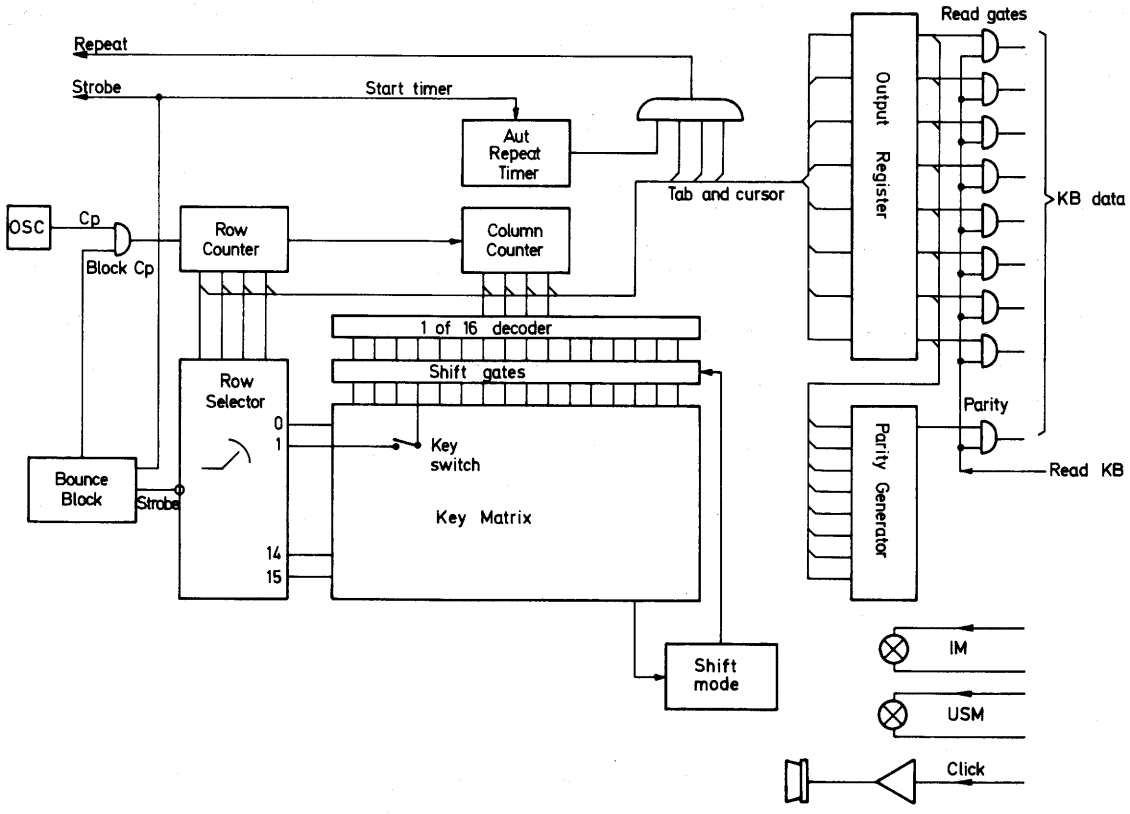


Figure 2. Block diagram of Keyboard

DETAILED DESCRIPTION

The Row Counter, dip 34, contains four flip-flops, which are connected to the Selector, dip 36. The Column Counter, dip 29, is decoded in dip 12, in order to provide the 16 columns. The Decoder outputs are connected to the Keyboard Matrix as well as the 16 row inputs of the Selector. The counters are clocked by the 50 kHz Cp, from dip 21. Each position 15 of the Row Counter, the Column Counter is stepped.

When the Selector senses a depressed key, a high level signal is applied from pin 10, which sets the Strobe FF 15/3 to ONE. This in turn triggers the schmittrigger formed as monostable FF. A 5 μs strobe pulse is now given to the STL. This also sets the monostable FF dip 10 during about 1 ms which in turn sets the Bounce Block FF 15/11 to ZERO for about 1 ms. This time depends on the RC circuit at 27/8 which provides Cp for the Bounce Block FF and the Strobe FF. The Strobe FF is thus set to ONE during the time bounces may occur in the key switch.

As long as a key is depressed the Strobe FF is set to ONE (but > 1 ms). In this state the FF blocks the clock pulses to the counters, which then stop in a state which is significant to the depressed key. The pulse from the Schmittrigger 27/6, sets the code from the counters into the Output Register, dip 25 and 32, and the PC-bit is generated in dip 19. The PC-generator senses bit 0 through 7, and adds one bit in the Output Register, bit 8, if the sum is even; odd parity. The code in the register is fetched by the read pulse from the Serial Transfer Logic, which opens the Interface gates dip 8, 13 and 14.

Certain keys provide different codes when shifted. By pressing the shift-key, the value changes from one column to another. When the shift-key is depressed, the Shift Mode FF 20/3 is set to ZERO. If then a double coded key is depressed its connection to the Decimal Decoder is changed from one output to another.

Consider the 2–3 pin at C2, 3. When the shift key is idle, column 2 is decoded through dip 7 and 4. If on the otherhand the shift key is depressed, the signal from the ONE-side of the Shift Mode FF gates column 3 to pin C2, 3 to the key at hand. The FF is set to ONE by releasing the shift-key.
If any of the move-cursor-key is depressed for more than 0.4 sec, the repeat line output is activated as long as the key is depressed. The retriggered monostable FF, dip 6 is used to set 20/11 after 0.4 sec. Only if the depressed key is a move-cursor-key it will cause automatic repetition, since these keys are decoded in 33/6.

One key holds a lamp, which may be lit by an order from the Controller. E.g. if the Computer wants to send a message to the Display the lamp may start flashing. When the operator wishes the message to be displayed, he presses the key with the lamp, thus notifying the Computer that he is ready to receive the message.

Another key with a lamp is the insert mode key. The lamp is lit when the MPU has been notified that insert mode is valid, i.e. the key has been pressed. Insert mode condition is valid until the ins-mode key is depressed once more.

When a key is depressed a "click" comes from the loudspeaker. The signal to the loudspeaker is fed from the Controller via the driver dip 35. The click appears when MPU has entered actual character into DIM and then set the "click"-bit in the command byte.

Identification Device

Contents

GENERAL 1

BRIEF OUTLINE 1

DETAILED DESCRIPTION 3

- Diode PC board 3
- Photo Transistors 4
- Visual hole detection 4
- Current modes 5
- Time pulse generator and RWM logic 5
- Fault FF, Bottom Indicator and ID available 6
- Read ID from RWM 7

SUMMARY 8

FIGURES

1. Example of ID card 1
2. Optical principle 2
3. Block diagram 2
4. Diode PC board 3
5. Sensing principle 4
6. Visual hole detection 5
7. Read card summary 8

GENERAL

In order not to give unauthorized persons access to the computer store via the keyboard, an identifier is provided. It contains an optical card reader, which can read an identification card. At an ID request from the computer, the device transfers the information on the card to the computer, which in turn checks the validity of the characters.

The card can hold up to 15 ID characters and one NUL character of seven data bits and one odd parity bit in the form of punched holes. It is possible to mask these holes by optically complementary filters in such a way that only the reader can sense them. Thus they are not visible. This, however, is an option and normally the holes are visual.
The card has one cut corner, indicating which way to insert it. The card is inserted in a slot in the reader, whereupon the characters are written one by one into a Read Write Memory, RWM. This is made during the insertion of the card while the punched holes pass an eight channel photo sensing array. When the card is completely inserted, this is detected by a nineth channel; bottom indication. When the card has been read, the characters have been stored in RWM and no errors have been detected, an indicator lamp is lit and a signal is applied to notify that ID data is available.

BRIEF OUTLINE

The data on the ID card is arranged in the following way, figure 1.

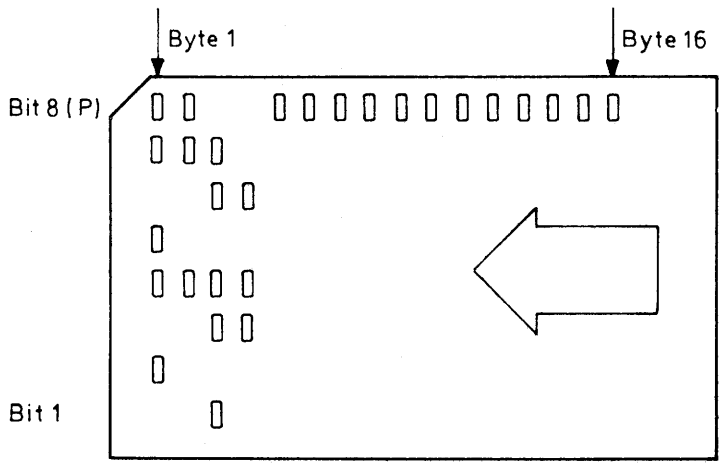


Figure 1. Example of ID card

The card is inserted in a slot. On one side of the slot there is a diode board. This is equipped with nine infrared light emmitting diodes, LED. In order to spare the LED they are controlled by three current mode control circuits. These circuits ensures that when the device is idle and no card is inserted, only three LED are lit with a low current rate. When the card is inserted all LED are lit until bottom is reached, and an idle state can be maintained again this time with all diodes turned off.

On the other side of the slot, directly opposit the LED, nine photo sensing circuits are assembled on a PC board, which also holds the rest of the logic.

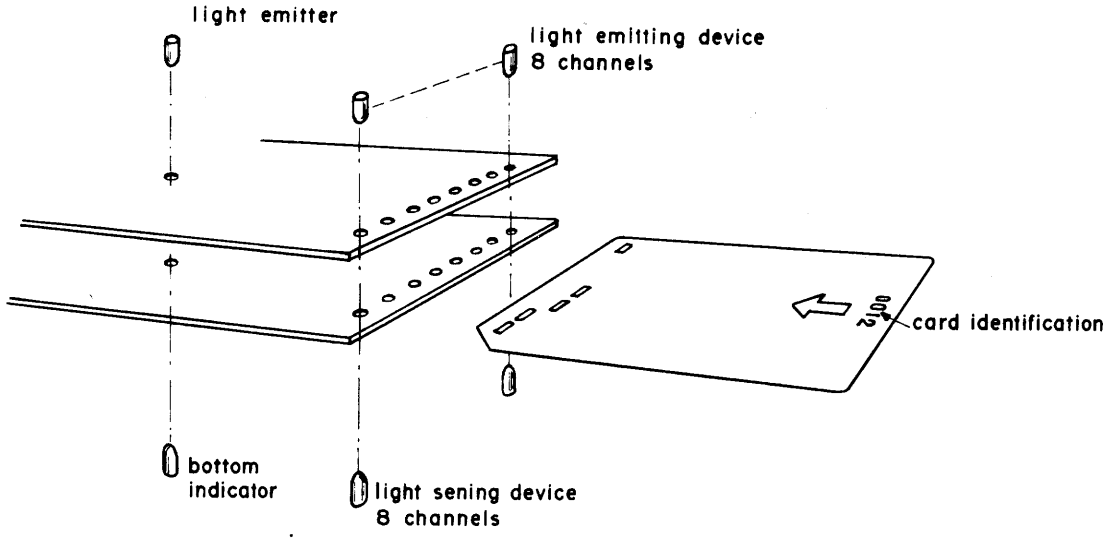


Figure 2. Optical principle

During the insertion of the card, the photo transistors are affected 16 times in different ways. Each time a "punched" character passes the eight data channels, the transistors sense light = ONE or dark = ZERO and the read character is fed in parallel to the RWM via a byte register.
Each character in the byte register is parity checked. When the option non visual holes is used, the logic detects any visual holes in the card and blocks the ID available signal to STL.

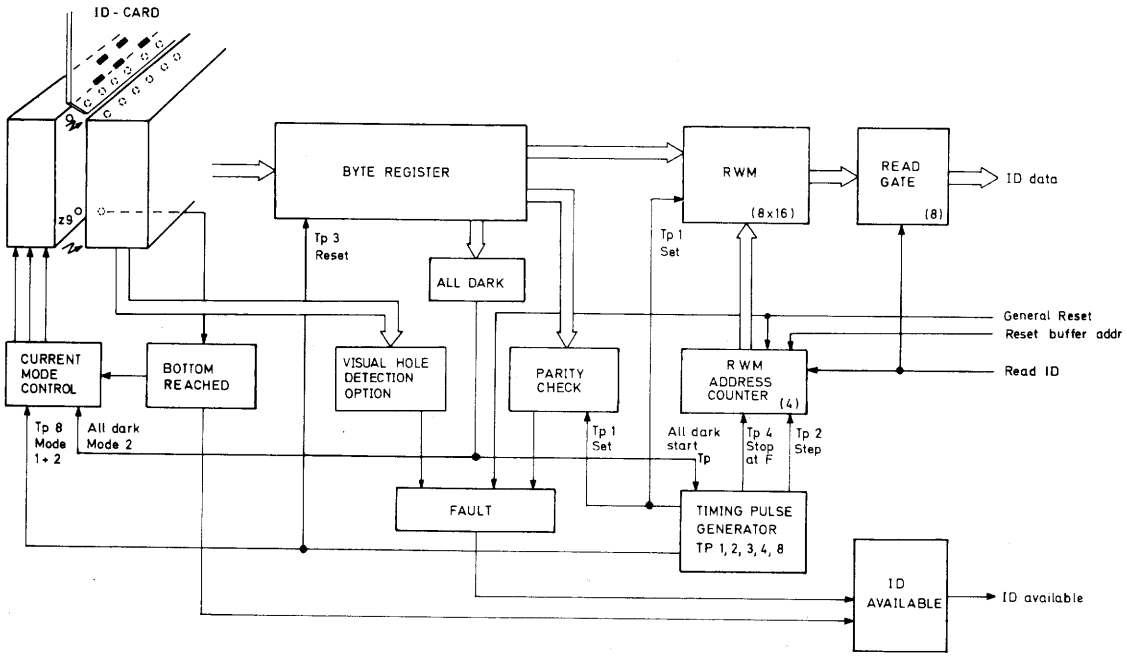


Figure 3. Block diagram

The RWM is controlled by a RWM address counter which is stepped for each detected character on the card.
When the card is fully inserted, the bottom indicator, channel 9, is covered by the card. At this state 16 characters are stored in RWM and provided that no error has been detected, the signal ID available is fed to STL.
The logic is, during the read phase described above, controlled from a Timing pulse generator, Tp generator, which completes a cycle of eight states between each detected character on the card. The Tp generator controls the current modes, provides load and reset condition for the byte register as well as write conditions and step pulses for RWM and its address counter. Finally the ending of the read phase at character 16 and "bottom reached" is controlled by the Tp generator.

Thus 16 ID characters are stored into RWM and ID available is fed to STL. When the computer requests an identification, e.g. at a transmission from a keyboard, STL feeds 16 Read ID pulses to the ID logic. These pulses steps the RWM address counter one cycle, and for each step an ID character is fetched from RWM output.

DETAILED DESCRIPTION

Diode PC board

The board contains the nine LED Z1–Z9. Diode Z9 is always lit and not affected by the current mode circuits.
In an idle state with no card inserted current mode 3 is valid. This means that Z7 and 8 (and Z9) are lit and a small current is fed via Y3 the base of which has low level. When a card is inserted it first covers channel 7. This is sensed by the logic and Y2 gets low level on its base and after a short time Y3 is blocked. All diodes are then lit and covered by the card and current mode 2 is at state. The diodes are still fed with a rather small current due to the value of R103.

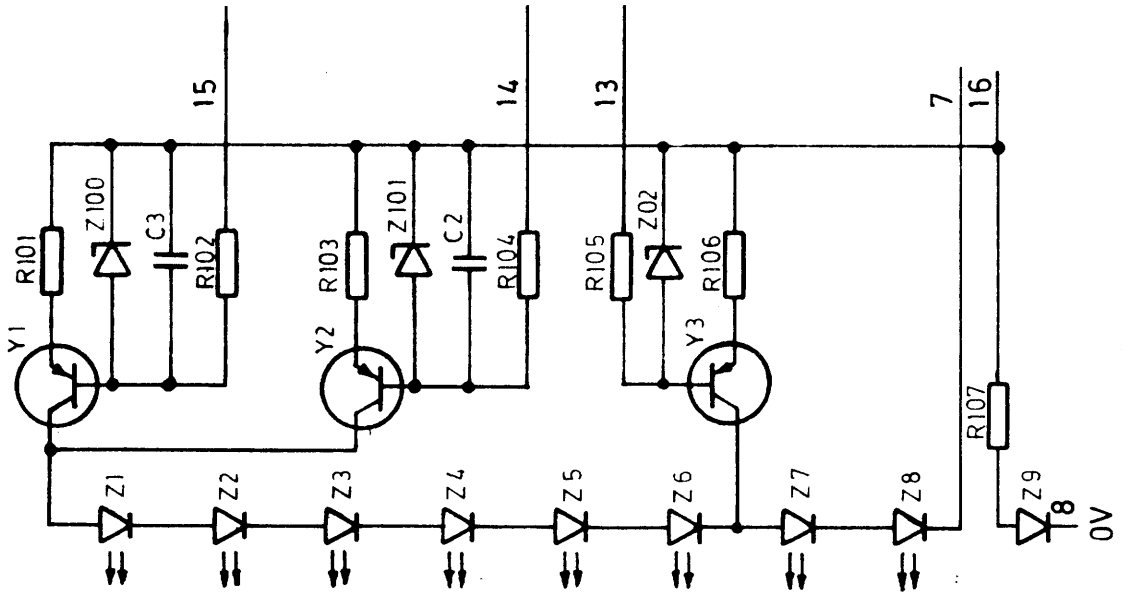


Figure 4. Diode PC board

When the punched holes of a character appear this is sensed by the logic and Y1 gets low level while Y2 is blocked. This is current mode 1 and due to the low value of R101 the current increases through the diodes for a short interval when the character on the card passes the diode array. The next instant the logic senses "all dark" and current mode 2 is valid again until the holes of next character are detected.
Thus the logic switches between mode 2 and 3 until Z9 is covered when all diodes are switched off except Z9. In this way the LED are spared since the increased current at mode 1 only appears during the reading. Otherwise this "read" current should burn out the diodes rather quickly.

Photo transistors

The array for reading the punched holes in the card contains nine equal circuits: 7 data, one PC and a bottom indicator. Consider channel 1 figure 5. The infrared light from Z1 is sensed by photo transistor Y12 the collector of which is connected to emitter follower Y11. Y12 conducts current as long as it senses the light and this implies that Y11 is blocked and the channel output at R13 feeds low level to the schmittrigger.

Low level means ONE and thus a hole has been detected. Should on the other hand Z1 be covered by the card, Y11 bottoms and a high level is fed from R13 to the schmittrigger, indicating ZERO or no hole.
Since odd PC is used, at least one channel must indicate ONE when a character position on a card passes the read array.

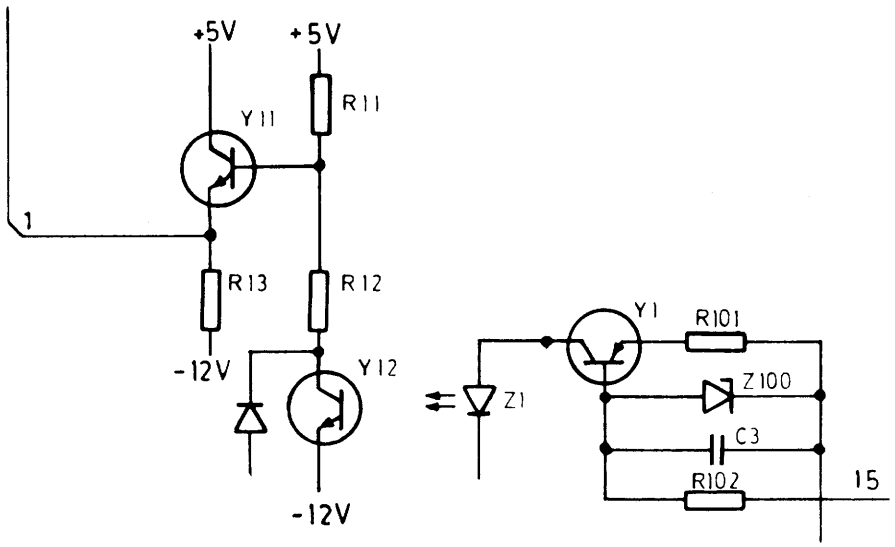


Figure 5. Sensing principle

Visual hole detection

If the option non visual holes is used the circuit Y5, Y4 is connected. Should in that case a card with visual holes be inserted, this is detected when the photo transistors bottoms. Normally shaded light is exposed to them via the optical filters covering the holes in the card. However, if at current mode 1 open light from the LED are applied, e.g. Z11 in channel 1 starts to lead current. Transistor Y5 thereby opens and transistor Y4 feeds a low level signal to schmittrigger 22/10, which sets

the error FF 11/3.
The signal ID available can then not be fed to DA.

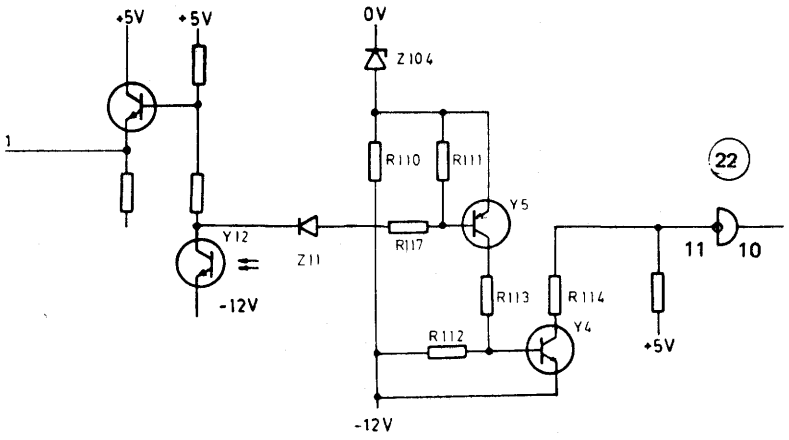


Figure 6. Visual hole detection

Current modes

At current mode 3 channels 7 and 8 feeds low level = ONE (no card inserted = hole). When the card is inserted the correct way channel 7 first is covered due to the cut corner of the card. Schmittrigger 24/4 thereby feeds low level to 15/11, 12. The card detected FF is reset and thus 14/3 opens Y2 for current mode 2. At this state both mode 2 and 3 is valid. An instant later channel 8 is covered and 17/8 feeds low level, indicating all dark. The card detected FF 3/3 is set from the Timing pulse generator, Tp generator. The signal from 3/5 sets FF 3/11. Card detected FF also blocks current mode 3. The gates 14/11 and 14/8 are now open for either current mode 2 or 1 since both 3/6 and 10/13 feeds low level to 10/10.
As long as all dark is valid at 17/8, mode 2 is maintained by the low level from 14/8 but when 17/8 rises to high level at the detection of a hole in the card, 14/11 opens for mode 1 and blocks mode 2 at 14/10. In this way mode 1 and 2 alternates during the insertion of the card.
When the card is completely inserted, schmittrigger 22/8 feeds low level to 10/11, 12 and thereby 10/10 blocks 14/11 and 14/8.

Time pulse generator and RWM logic

The Tp generator contains one eight bits shift register, IC6, the start FF 7/3, and the oscillator 22/2. When the generator is idle 7/3 is set and ONE's are shifted into the register at about 1 MHz from 22/2. When 17/8 detects all dark, 7/3 is reset and at the next clock pulse from 22/2 a ZERO is set in IC6. The ZERO immediately appears at 6/3 and 7/3 is set again. Thus only one ZERO is shifted through the shift register, indicating the eight Tp.

The first Tp cycle implies

Tp1	set 7/3 (START FF)
Tp3	reset byte register 8, 12, 16, 21

Tp8 sets 3/3, condition for current mode 1 or 2

When the first ID character is sensed, i.e. a hole is detected, current mode 1 provides light for safe scanning of the character.

The character from the card is set into the byte register and is parity checked in IC20 and 18/8. The next instant the Tp generator is started by all dark from 17/8 and this cycle implies:

Tp1 set 2/3, load RWM from byte register via 18/6 and open 19/6 to enable setting of fault FF

Tp2 step RWM address counter IC13 from idle state 1111 to 0000

Tp3 reset byte register

The lap is repeated for each character and each Tp4 the gate 15/4 is checked regarding the RWM address. This is to detect when address 1111 appears, i.e. when all characters are entered into RWM.

The RWM logic mainly contains two 4 x 16 bits RWM and a four bits address counter. The idle state of the counter is 1111. When a card is inserted into the reader, the Tp generator is started at the first all dark.
At Tp2 a step pulse is fed to 15/3, but this gate is blocked since 3/11 is reset and will not be set until the first Tp8.
Thus when the first character is set into the byte register the counter is blocked. This implies that the next time all dark is decoded, Tp1 sets the first character into RWM address 1111 via 18/6 and 19/3. Tp2 then steps the address counter to 0000, which is the address for the second character. This is possible since 3/11 has been set at the previous Tp8. Tp4 then opens 15/4 but the clear condition to 3/11 has now disappeared from 17/6 since the counter is in state 0000.

When all 16 characters has been read into RWM, the counter has reached 1110 and Tp1 has appeard for the last time. Tp2 steps the counter again and 1111 is fed to RWM, providing the first address for a possible Read ID from STL. The decoder gate 17/6 now feeds low level again and when the last Tp4 appears 3/11 is cleared and blocked. The next instant the bottom indicator is covered by the card and schmittrigger 22/8 feeds low level to 3/12.

Fault FF, Bottom Indication and ID available

Each character in the byte register is parity checked in IC20 and 18/8. If parity bit 8 does not agree with the parity from IC20, 18/8 feeds high level and at the next Tp1 the fault FF 11/3 is set. This in turn sets the ID available FF 11/11 and the signal ID available thereby gets high = not valid.

If the non visual hole option is used, the fault FF is set at the detection of a visual hole. Schmitttrigger 22/10 (figure 6) sets the fault FF, since 3/3 is set during the read phase, and 11/2 gets high level. The ID available FF is thus blocked in ONE state both at parity fault and at the detection of a visual hole.

When the ID card is completely inserted, channel 9 is covered. Schmittrigger 22/8 then feeds low level which sets 7/11 when 3/11 is cleared at the last Tp4. This indicates bottom reached. If the fault FF has not been set at this stage, the ID-available FF will be zero-set. When 11/11 is zero-set the indicator LED on the panel is lit. Since 3/11 is reset the address counter cannot be stepped

by Tp2 until the ID card is pulled out and inserted again for a new reading. Since bottom is reached, 10/13 blocks current mode 1 and 2 in 10/10, i.e. all diodes except diode 9 are put out.

Note that the Cp to the bottom reached FF, 7/11, only appears once. When 17/6 decodes 1111, 3/11 is reset and thus 3/8 feeds a Cp to 7/11. This means that the bottom reached FF only can be set when 16 characters has been counted.

When the card is removed, 22/8 feeds high level to 10/11, 12. Now the condition for Cm2 and 1 is valid.
When the card is out, all photo transistors sense light and the condition "all light" is valid for a moment. 23/8 clears 3/3 and now Cm3 is valid.

Read ID from RWM

When 11/11 is reset the low level ID available is fed to STL. STL then feeds 16 Read ID pulses. Each pulse opens the read gates 1 and 4 and steps the address counter once. Should ID available have high level when Read ID appears, NUL characters are applied with correct parity from the read gates. The pulses appear every nine sweep of the display, since each bit needs one sweep time for transfer from STL to DA, and STL adds a nineth bit before the transfer.

SUMMARY

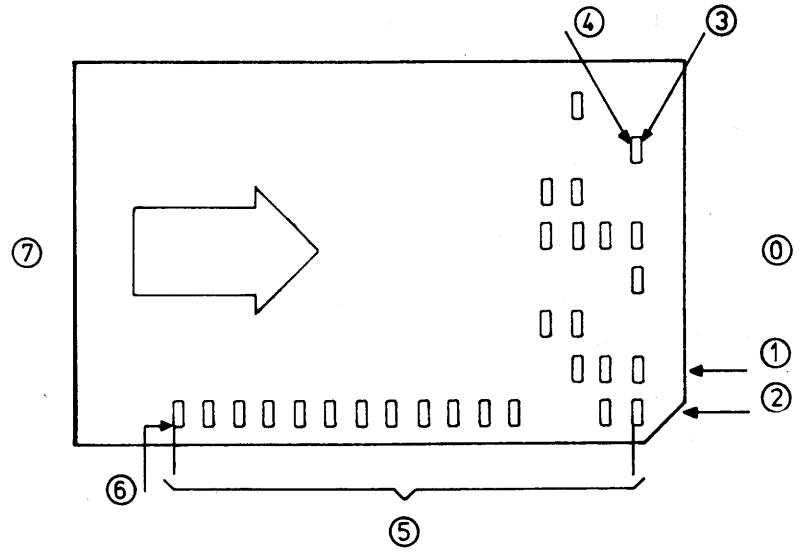


Figure 7. Read card summary

0 Card out. 3/3 has been cleared from 23/8. 3/5 maintains Cm3.

1 Channel 7 senses edge of card. Cm2 + 3 is valid, Cm2 via 15/13 and 3/6.

2 Channel 8 is covered. Tp-generator is started:

All dark sets 7/3 to ZERO

Tp1 presets 7/3

Tp2 is blocked in 15/1

Tp3 resets byte register

Tp8 sets 3/3 which sets 3/11 and gives condition for Cm2 + 1 via 10/10. Cm3 is blocked.

3 First byte is sensed. 17/8 rises to high level and opens 14/11 for Cm1.

4 All dark blocks 14/11; 14/9 opens for Cm2. Tp-generator is started.

Tp1 sets 7/3, opens 19/6 for parity check and sets character from byte register into RWM

Tp2 steps address counter via 15/11

Tp3 resets byte register

Tp4 opens 15/4 to check if address counter has reached 1111

5 3 and 4 repeated

6 Byte 16 is sensed and set into byte register. Address counter has reached 1110. All dark appears and bottom indicator (Z9) is covered.

Tp1 sets 7/3, checks PC and sets byte 16 into RWM

Tp2 steps address counter to idle state 1111

Tp3 resets byte register

Tp4 checks 17/6 in 15/4. Thereby 3/11 is reset. 3/8 blocks further counting by Tp2 and provides Cp for 7/11.

When Z9 is covered Cm2 + 1 are blocked from 10/13. Only Z9 is lit.

7 If no error has been made during insertion, 11/11 can be set to ZERO. This opens: 9/13 for address counting and IC4 and 1 for reading from RWM.

If 11/11 is set (error, no card), 9/13 and 4 and 1 are blocked. Read pulses thereby generates PC-bit = ONE in 9/4 and NUL with correct parity is applied from the read gates.

8 Card out. All light clears 3/3. 3/5 maintains Cm3.

STANSAAB

Alfaskop 3500
Data Terminal System

Drawings and Diagrams

EE567–412

Control Unit 3501
- 1. Contents, Cable run sheets, PS
- 2. MUX, TIM, DA, PA
- 3. MPU, MEM
- 4. MCA, EBA, BUT, LM

Control Unit 3502
- 5. Contents, Cable run sheets
- 6. COB, CHI

Display Unit 3510
- 7. Contents, Cable run sheets, CRT unit, STL
- 8. KB, ID, LPA

Display Station 3550
- 9. Contents, Cable run sheets, DIL, DKA

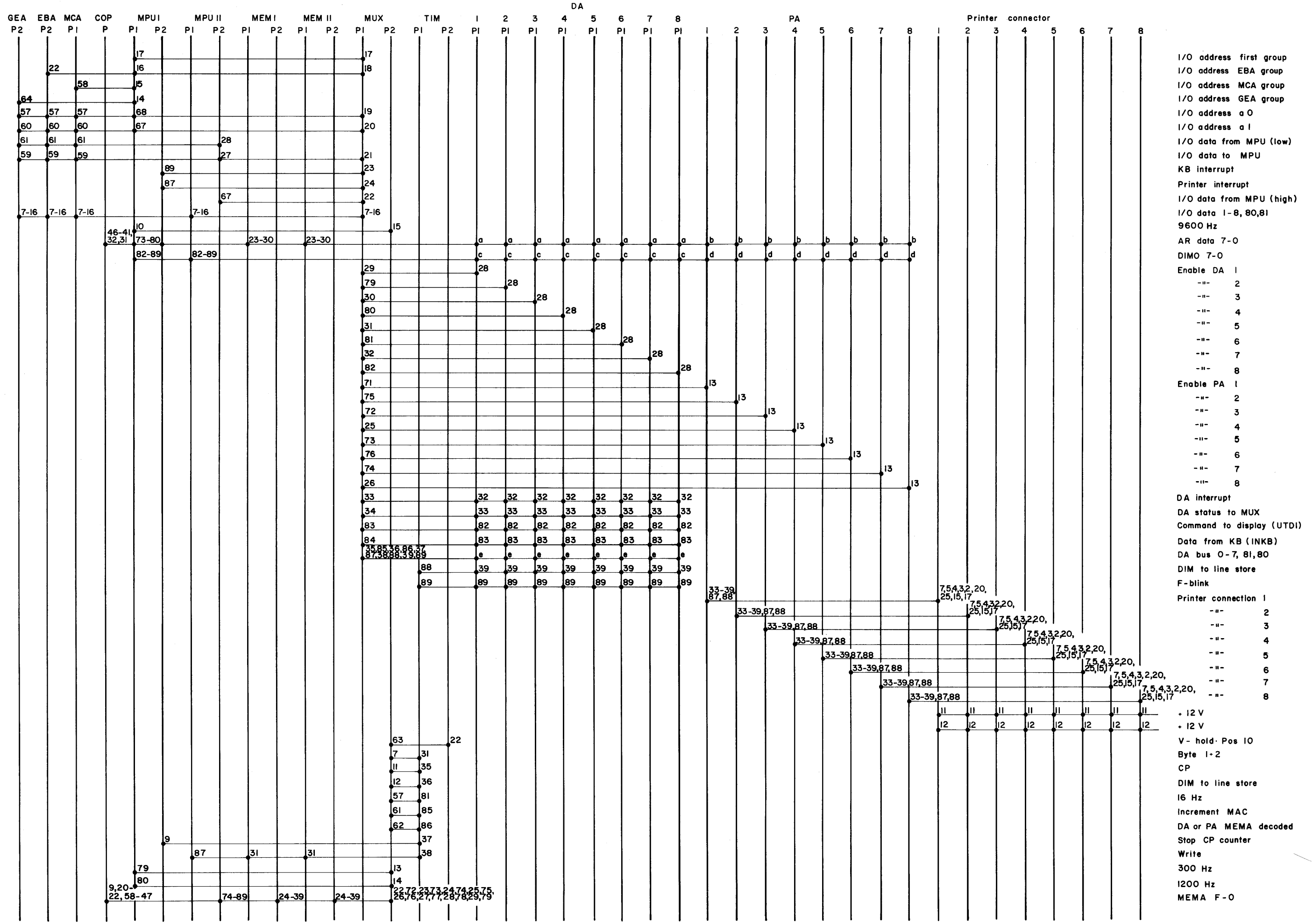
Special options
- 10. Contents

DRAWINGS AND DIAGRAMS

Control Unit 3501

Part 1	3501 Interconnections		4 sheets
	PS,	Power Supply	
		Power distribution	E27413 2001 8
		Circuit diagram	E27404 2000 2
Part 2	MUX,	Multiplexor	
		Reference sheet	E07150 2090 3, sh 1
		Block diagram	E07150 2090 2, sh 2
		Logic diagram	E07150 2090 1, sh 3–4
		PCB assembly	E07150 0090 3, sh 1, 3
	TIM,	Timer	
		Reference sheet	E07150 2100 3, sh 1
		Block diagram	see MUX
		Logic diagram	E07150 2100 1, sh 3–4
		PCB assembly	E07150 0100 3, sh 1, 3
	DA,	Display Adapter	
		Reference sheet	E07150 2110 3, sh 1
		Block diagram	E07150 2110 3, sh 2
		Logic diagram	E07150 2110 1, sh 3–5
		PCB assembly	E07150 0110 3, sh 1, 3
	PA,	Printer Adapter	
		Reference sheet	E07150 2120 3, sh 1
		Block diagram	E07150 2120 3, sh 2
		Logic diagram	E07150 2120 7, sh 3a, 3b
		PCB assembly	E07150 0120 3, sh 1, 3
Part 3	MPU,	Micro Processing Unit	
		MPU I Reference sheet	E07150 2000 3, sh 1
		Block diagram	E07150 2000 2, sh 2
		Logic diagram	E07150 2000 1, sh 3–4
		PCB assembly	E07150 0000 3
		MPU II Reference sheet	E07150 2010 3, sh 1
		Block diagram	See MPU I
		Logic diagram	E07150 2010 1, sh 2–3
		PCB assembly	E07150 0010 3

	MEM,	Memories	
		MEM 6/2 Reference sheet	E07150 2130 3, sh 1
		Block diagram	E07150 2130 3, sh 2
		Logic diagram	E07150 2130 1, sh 3
		PCB assembly	E07150 0130 3
		MEM 4/0 Reference sheet	E07150 2131 3, sh 1
		Block diagram	E07150 2131 4, sh 2
		Logic diagram	E07150 2131 1, sh 3
		PCB assembly	E07150 0131 3, sh 1
		MEM 4/4 Reference sheet	E07150 2020 3, sh 1
		Block diagram	E07150 2020 3, sh 2
		Logic diagram	E07150 2020 1, sh 3
		PCB assembly	E07150 0020 3
		MEM 4/0 Reference sheet	E07150 2021 3, sh 1
		Block diagram	E07150 2021 4, sh 2
		Logic diagram	E07150 2021 1, sh 3
		PCB assembly	E07150 0021 3
		MEM 4/4 Reference sheet	
		Block diagram	See E07150 2020
		Logic diagram	
		PCB assembly	E07150 0022 3
Part 4	MCA,	Modem Communication Adapter	
		Reference sheet	E07150 2050 3, sh 1
		Block diagram	E07150 2050 2, sh 2
		Logic diagram	E07150 2050 7, sh 3–4
		PCB assembly	E07150 0050 3, sh 1
	EBA,	External Bus Adapter	
		Reference sheet	E07150 2070 3, sh 1
		Block diagram	E07150 2070 2, sh 2
		Logic diagram	E07150 2070 1, sh 3
		PCB assembly	E07150 0070 3
	BUT,	Bus Terminator	E28012 0001 4
	LM,	Line Multiplexor	
		Circuit diagram	E27429 2000 7



a = 20,70,21,71,22,72,23,73
b = 17, 67,18,68,19,69, 20,70
c = 24, 74,25,75,26,76, 27, 77
d = 21, 71, 22,72,23,73, 24,74
e = 34,84,35,85,36,86,37,87,38,88

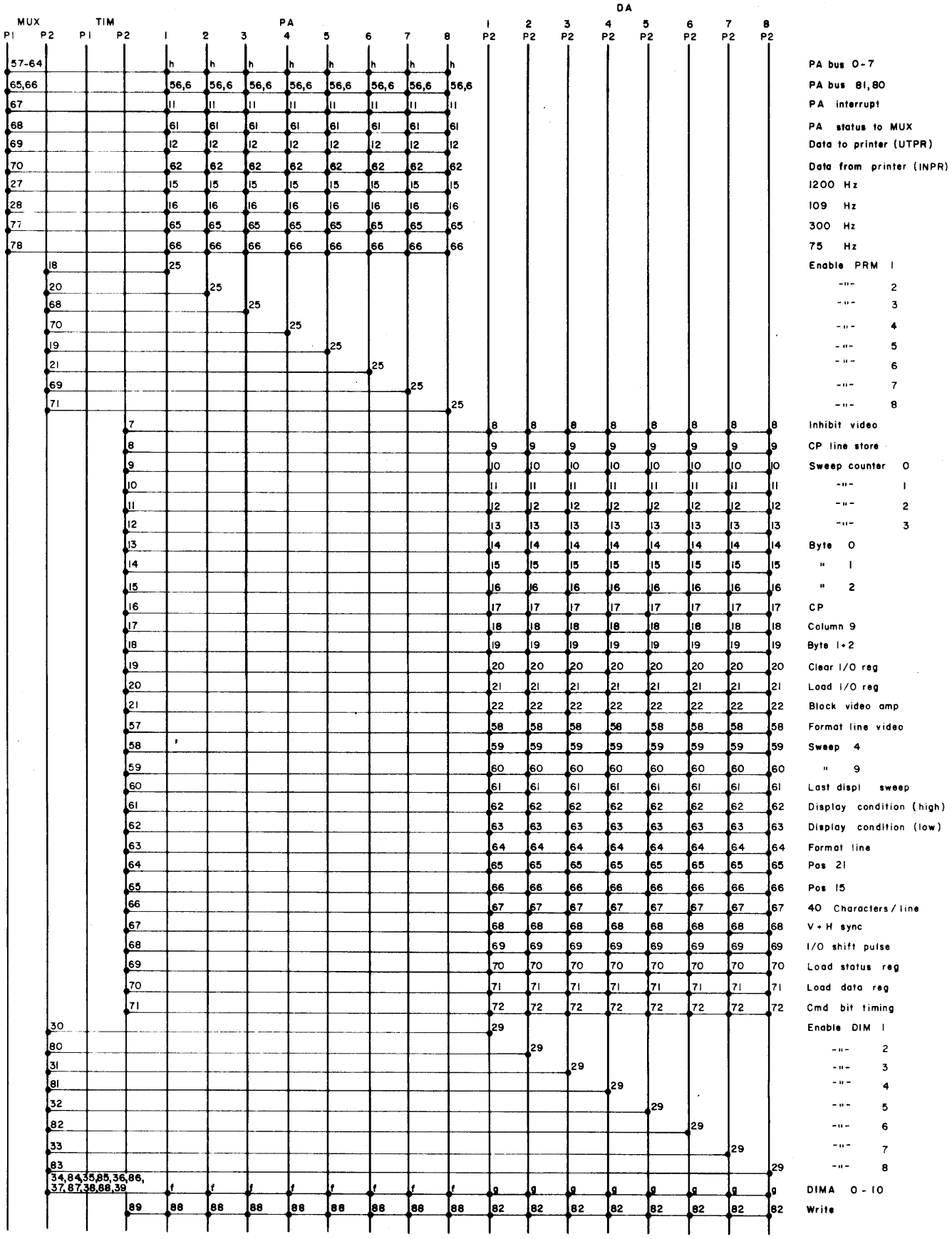
Control Unit 3501 Interconnection
PC-board

This document must not be copied, reprinted or disclosed to any third party, either wholly or in part, without our consent. Any infringement will lead to legal proceedings.
STANSAAB Elektronik AB SWEDEN

OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!

Denna handling får ej utan vårt medgivande bekantgöras kopieras mångfaldigas eller eljest obehörigen utnyttjas. Överträdelse beivras med stöd av gällande lag.
STANSAAB Elektronik AB

REV. 1



PA bus 0-7
PA bus 81, 80
PA interrupt
PA status to MUX
Data to printer (UTPR)
Data from printer (INPR)
1200 Hz
109 Hz
300 Hz
75 Hz
Enable PRM 1
-"- 2
-"- 3
-"- 4
-"- 5
-"- 6
-"- 7
-"- 8
Inhibit video
CP line store
Sweep counter 0
-"- 1
-"- 2
-"- 3
Byte 0
" 1
" 2
CP
Column 9
Byte 1+2
Clear I/O reg
Load I/O reg
Block video amp
Format line video
Sweep 4
" 9
Last displ sweep
Display condition (high)
Display condition (low)
Format line
Pos 21
Pos 15
40 Characters / line
V + H sync
I/O shift pulse
Load status reg
Load data reg
Cmd bit timing
Enable DIM 1
-"- 2
-"- 3
-"- 4
-"- 5
-"- 6
-"- 7
-"- 8
DIMA 0 - 10
Write

f = 27, 77, 28, 78, 29, 79, 30, 80, 31, 81, 32, 82

g = 33, 83, 34, 84, 35, 85, 36, 86, 37, 87, 38, 88

h = 7, 57, 8, 58, 9, 59, 10, 60

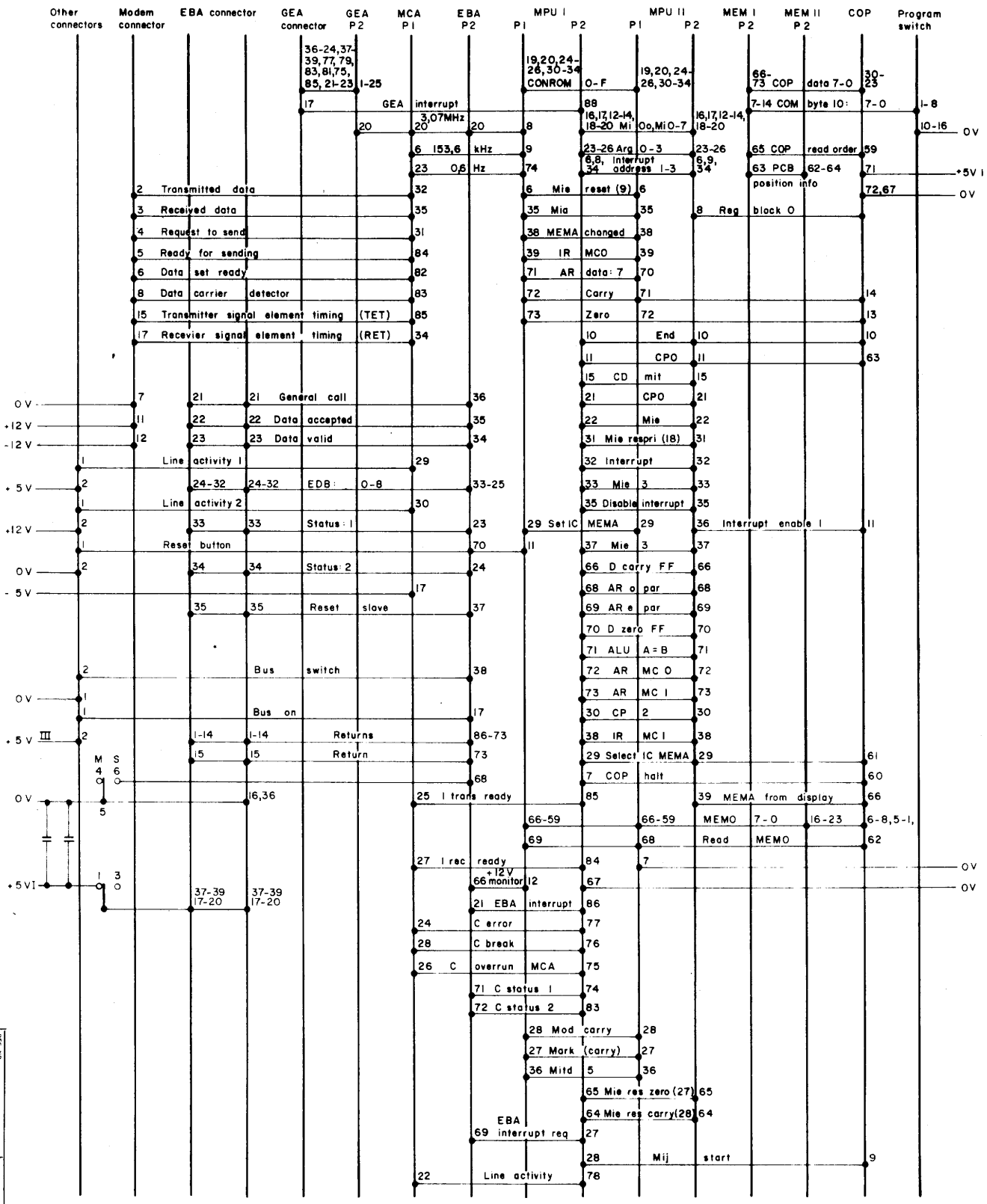
Control Unit 3501 Interconnection
PC-board

STANSAAB

This document must not be copied, reprinted or disclosed to any third party, either wholly or in part, without our consent. Any infringement will lead to legal proceedings.
STANSAAB Elektronik AB SWEDEN

OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!

Denna handling får ej utan vårt medgivande bekantgöras kopieras mångfaldigas eller eljest obehörigen utnyttjas. Överträdelse beivras med stöd av gällande lag.
STANSAAB Elektronik AB

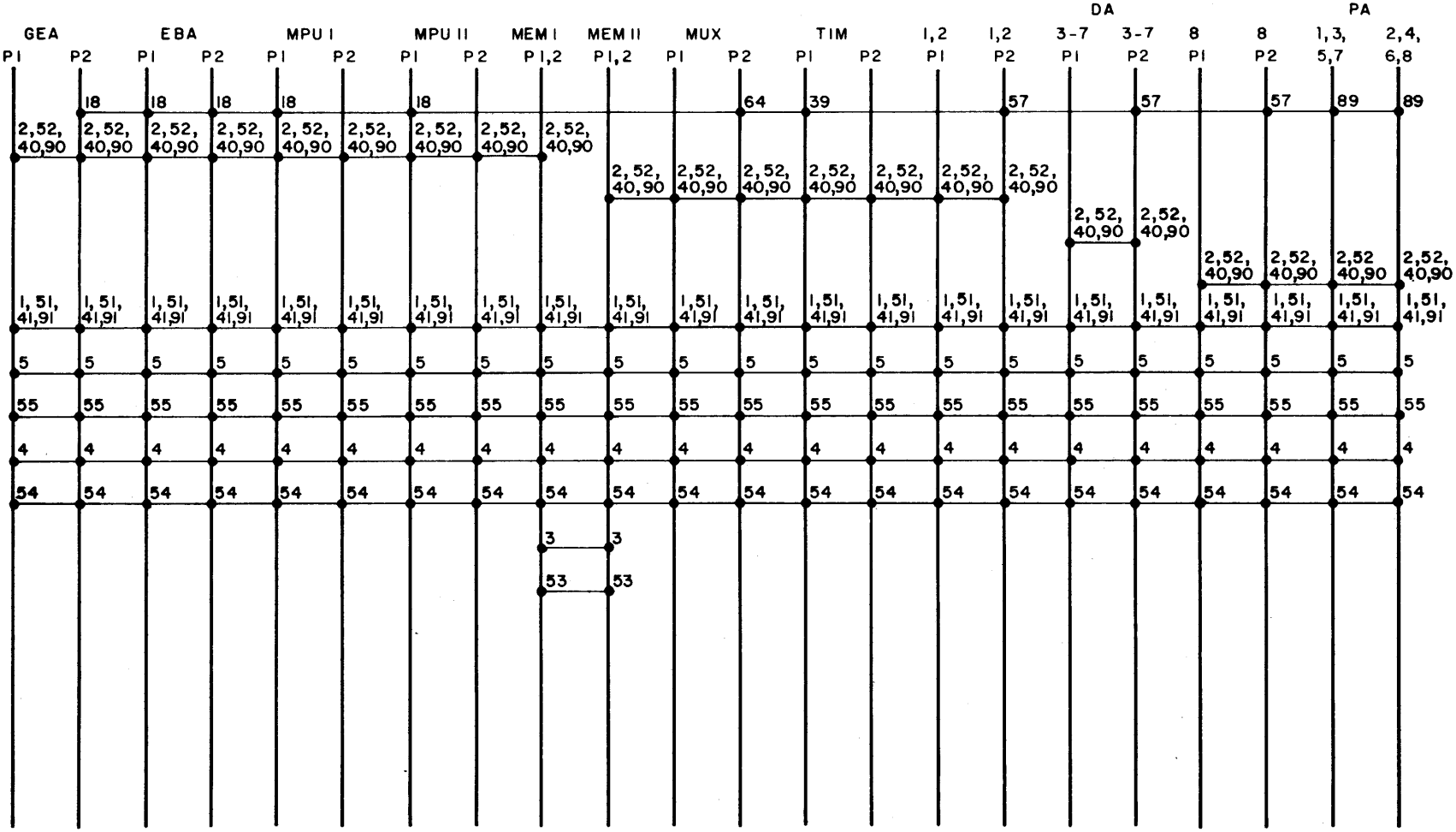


Control Unit 3501 Interconnection
PC-board

STANSAAB

Denna handling får ej utan vårt medgivande bekantgöras kopieras mångfaldigas eller eljest obehörigen utnyttjas. Överträdelse beivras med stöd av gällande lag.
STANSAAB Elektronik AB

OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!



Control Unit 3501 Interconnection
PC-board

This document must not be copied, reprinted or disclosed to any third party, either wholly or in part, without our consent. Any infringement will lead to legal proceedings.
STANSAAB Elektronik AB SWEDEN

INGÅR ÄVEN I / ALSO USED ON			REG.-NR / REG. No
KONTR / CHECK		GODK / APPROV	
SKALA / SCALE		KOPIER / COPIED	
UTFÖRD / ISSUED			

STANSAAB

BLAD / SHEET

REV.1

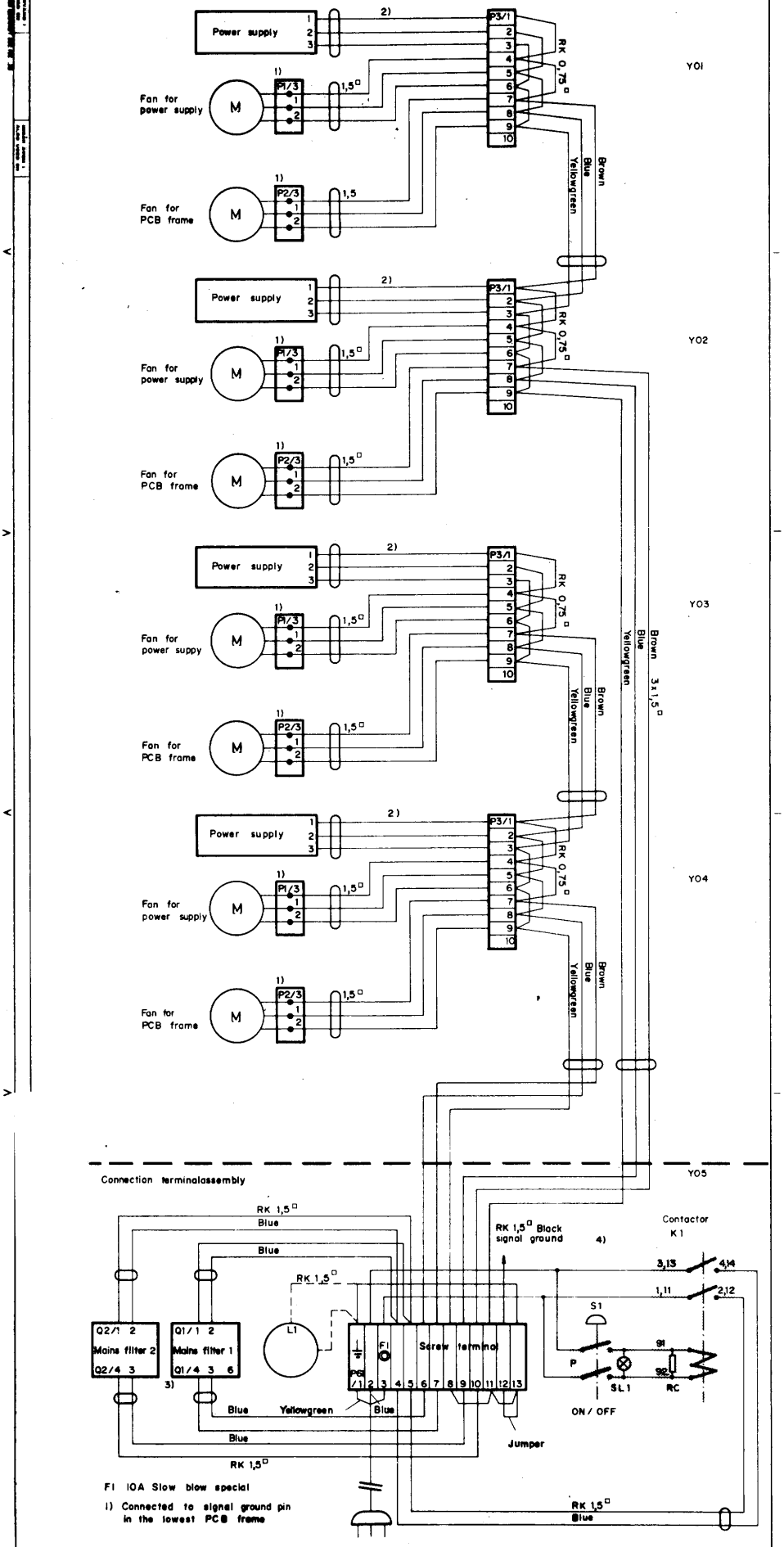
FÖRST ANVÄND I / FIRST USED ON

INGÅR ÄVEN I / ALSO USED ON

This document must not be copied, reprinted or disclosed to any third party, either wholly or in part, without our consent. Any infringement will lead to legal proceedings.
STANSAAB Elektronik AB SWEDEN

OM NAGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!

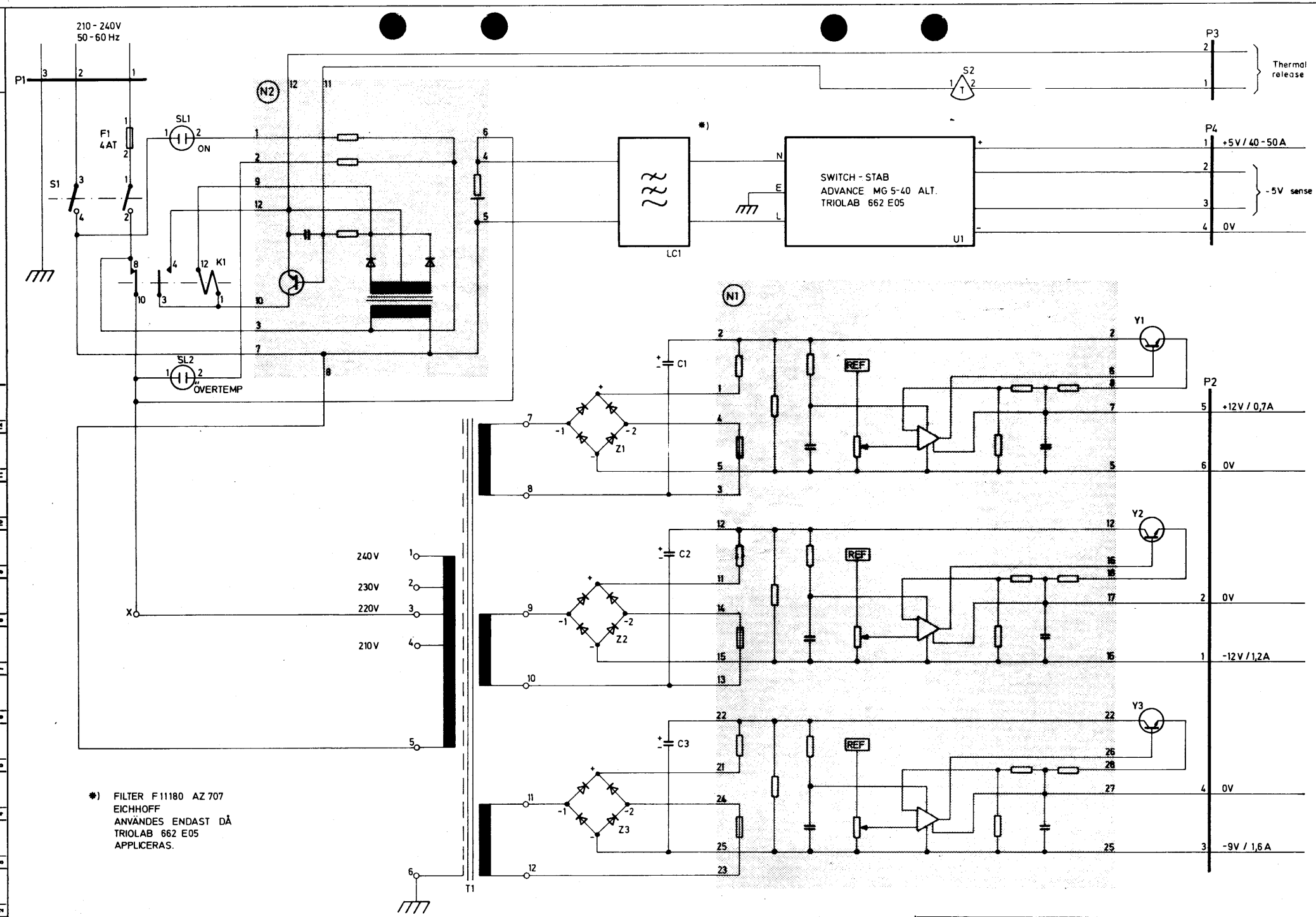
Denna handling får ej utan vårt medgivande bekantgöras kopieras mångfaldigas eller eljest obehörigen utnyttjas. Överträdelse beivras med stöd av gällande lag.
STANSAAB Elektronik AB



REG. NR / REG. NO			
KONTR. / CHECK		GODK. / APPROV.	
SKALA / SCALE		KOPIER / COPIED	8506 9.5
UTFÄRD / ISSUED	8929 Wed	74 03 19	

Power distribution
3501 , 3502

STANSAAB
E 27413 2001 8E



REG.-NR / REG. NO			
KONTR / CHECK		GODK / APPROV	
SKALA / SCALE		KOPIER / COPIED	8506 GS
UTFÄRD / ISSUED	974 Gbi		74 01 18

Power supply circuit diagram

STANSAAB

E 27404 2000 2 E

REV. 1

FÖRST ANVÄND I / FIRST USED ON

INGÅR ÄVEN I / ALSO USED ON

OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!
STANSAAB Elektronik AB SWEDEN

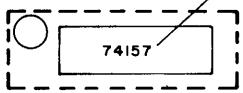
TEST POINTS

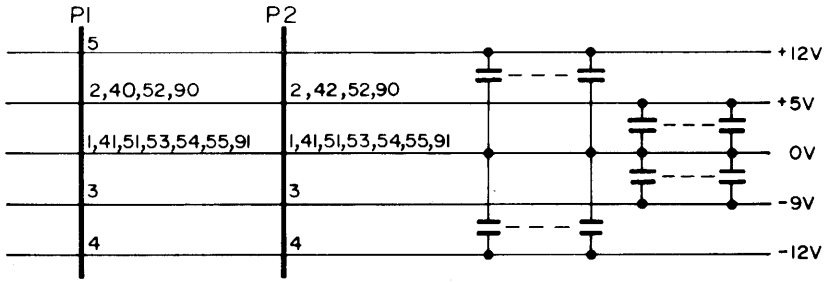
NR	SIGNAL
1	KB interrupt
2	
3	
4	INPA
5	PA address bit 0
6	Printer interrupt
7	PA interrupt
8	PA address bit 2
9	
10	
11	
12	PA address bit 1
13	Output mode PA
14	
15	PA irpt FF
16	
17	UTPR
18	UTPA
19	INDA
20	UTDI
21	UTDA
22	
23	
24	
25	
26	
27	
28	
29	
30	INKB
31	Output mode DA
32	DA irpt FF
33	DA address bit 2
34	DA address bit 1
35	DA address bit 0
36	Byte 1+2
37	CP 0
38	MAC carry 1
39	Enable DA memory
40	Enable PA memory
41	MAC carry 2



2N 2907

INDICATED NUMBERS ONLY
EXAMPLES OF EXISTING
CIRCUIT TYPES





REVISION OF DIAGRAMS

DIAGRAM AND DRAWING NUMBER	SH.	STATUS MOD.
E 07150 2090 3	1	REV.
E 07150 2090 2	2	"
E 07150 2090 1	3	"
E 07150 2090 1	4	"
		"
		"
		"
		"
		"
		"

INGÅR AVEN I
ALSO USED ON

FÖRST ANVAND
FIRST USED ON

KONTR CHECK
SKALA SCALE
UTFÖRD ISSUED 8427 LA

GODK APPROV. 0226 LB
KOPIER COPIED 8506 G.S.
73 11 05

MUX
Multiplexor

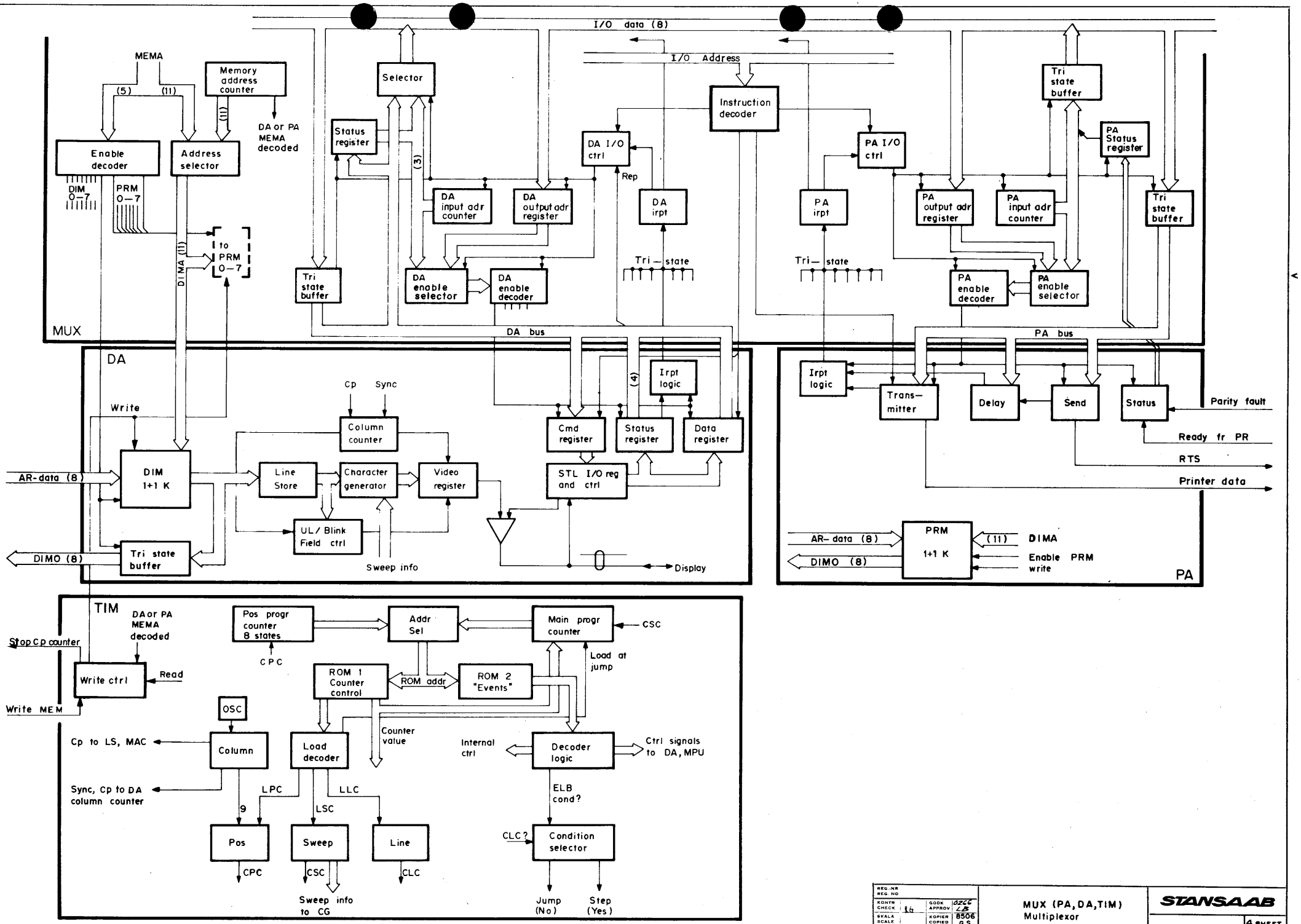
REG.-NR
REG. No

STANSAAB

E 07150 2090 3

4 SHEET 1

MUX
REV.I



Denna handling får ej, utan vårt medgivande behandlas kopieras mångfaldigas eller eljest obehörigen utnyttjas. Överträdelse beivras med stöd av gällande lag
STANSAAB Elektronik AB

OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!

This document must not be copied, reprinted or disclosed to a third party without our consent. Any infringement will lead to legal proceedings
STANSAAB Elektronik AB SWEDEN

REG.NR / REG. NO			
KONTR CHECK	LA	GODK APPROV	AZCC LB
SKALA SCALE		KOPIER COPIED	8506 G.S.
UTFÖRD ISSUED	R474 RH	74 08 13	

MUX (PA, DA, TIM)
Multiplexor

STANSAAB

E 07150 2090 2

4 SHEET 2

REV.1

FÖRST ANVÄND I: FIRST USED ON

INGÅR ÄVEN I: ALSO USED ON

Memory address counter (MAC)

Memory address - selector

Memory enable decoder

Enable DA memory

Enable PA memory

Clock divider

Instruction decoder

PA status register

PA input address counter

PA output address register

PA address selector

PA enable decoder

Stop FF

Output desired

Stop

Input finished

Read PA status FF

PA irpt FF

TIM V-hold: Pos 10

TIM Increment MAC

MPU II MEMA: 0 … 15

TIM DIM to line store

MPU I I/O address EB/A group

PA PA interrupt

MPU I General reset

1/sh4

2/sh4

3/sh4

4/sh4 CP

5/sh4 GR

6/sh4 GR

7/sh3 I/O data

62 DA or PA MEMA decoded → TIM

18 Enable PRM 1 → PA 1
20 " 2 → " 2
68 " 3 → " 3
70 " 4 → " 4
19 " 5 → " 5
21 " 6 → " 6
69 " 7 → " 7
71 " 8 → " 8

30 Enable DIM 1 → DA 1
80 " 2 → " 2
31 " 3 → " 3
81 " 4 → " 4
32 " 5 → " 5
82 " 6 → " 6
33 " 7 → " 7
83 " 8 → " 8

34 DIMA: 0 → PA, DA
84 " 1 → "
35 " 2 → "
85 " 3 → "
36 " 4 → "
86 " 5 → "
37 " 6 → "
87 " 7 → "
38 " 8 → "
88 " 9 → "
39 " 10 → "

13 300 Hz → MPU I
14 1200 Hz → "

77 (300 Hz) C → PA
27 (1200 Hz) B → "
28 (109 Hz) A → "
78 75 Hz → "

64 PA bus 7 → PA
63 " 6 → "
62 " 5 → "
61 " 4 → "
60 " 3 → "
59 " 2 → "
58 " 1 → "
57 " 0 → "
66 " 80 → "
65 " 81 → "

26 Enable PA 8 → PA
74 " 7 → "
76 " 6 → "
73 " 5 → "
25 " 4 → "
72 " 3 → "
75 " 2 → "
71 " 1 → "

24 Printer interrupt → MPU I
68 PA status MUX → DA
69 Data to printer (UTPR) → "
70 Data from printer (INPR) → "

SIA

OM NÅGOT ÄR OKLART: FRÅGA! IF IN DOUBT: ASK!

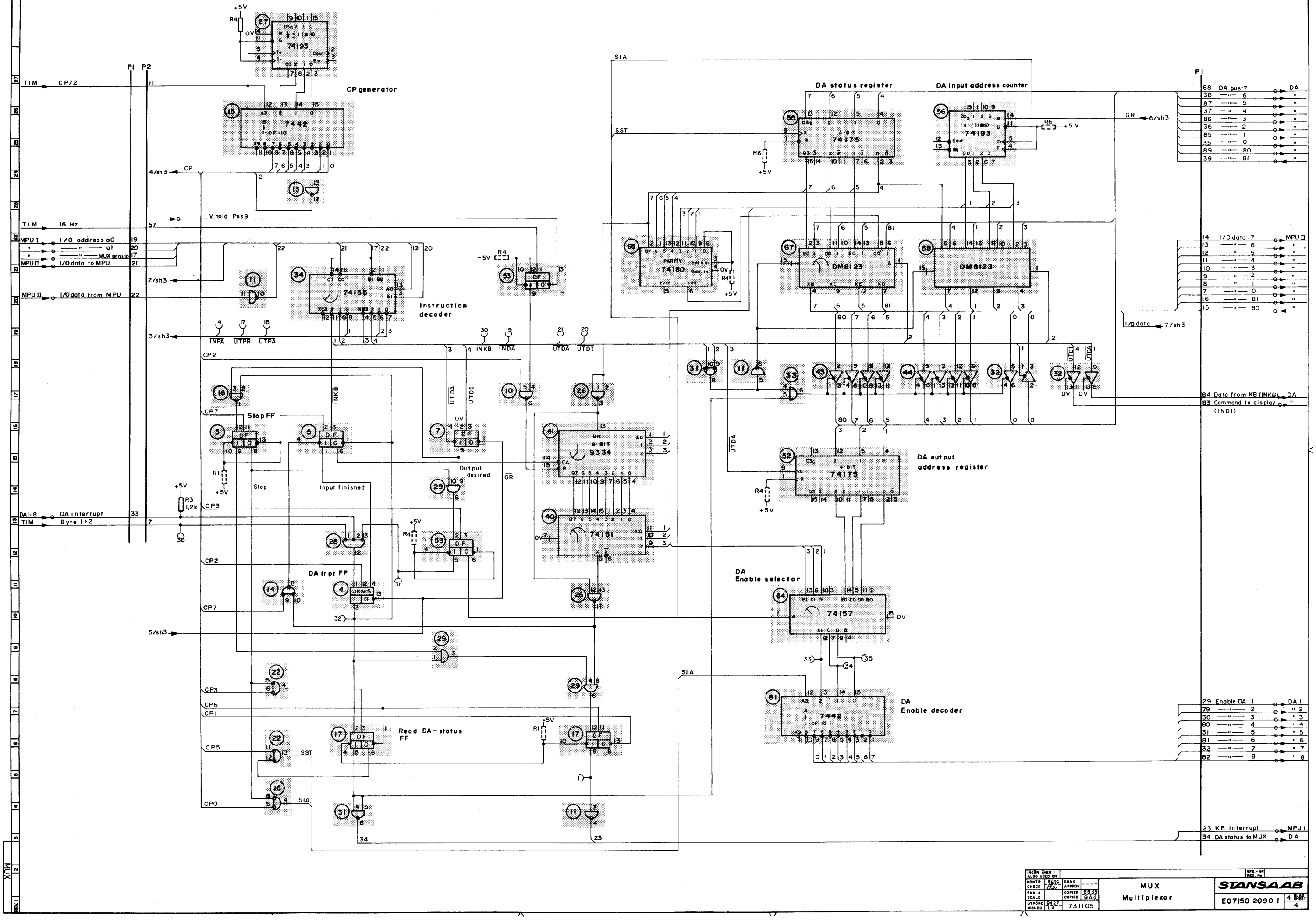
This document must not be copied, reprinted or disclosed to any third party, either wholly or in part, without our consent. Infringements will be prosecuted. STANSAAB assumes no responsibility.

Denna handling får ej utan vårt medgivande kopieras, återges, utlämnas eller på annat sätt användas av obehörig. Överträdelse beivras. STANSAAB ansvarar ej för tillämpningar.

INGÅR ÄVEN I ALSO USED ON			REG.-NR REG. NO
KONTR CHECK	R429 26/2	GODK APPROV	----
SKALA SCALE		KOPIER COPIED	8839 B.A.O.
UTFÄRD ISSUED	84 27 LA	731105	

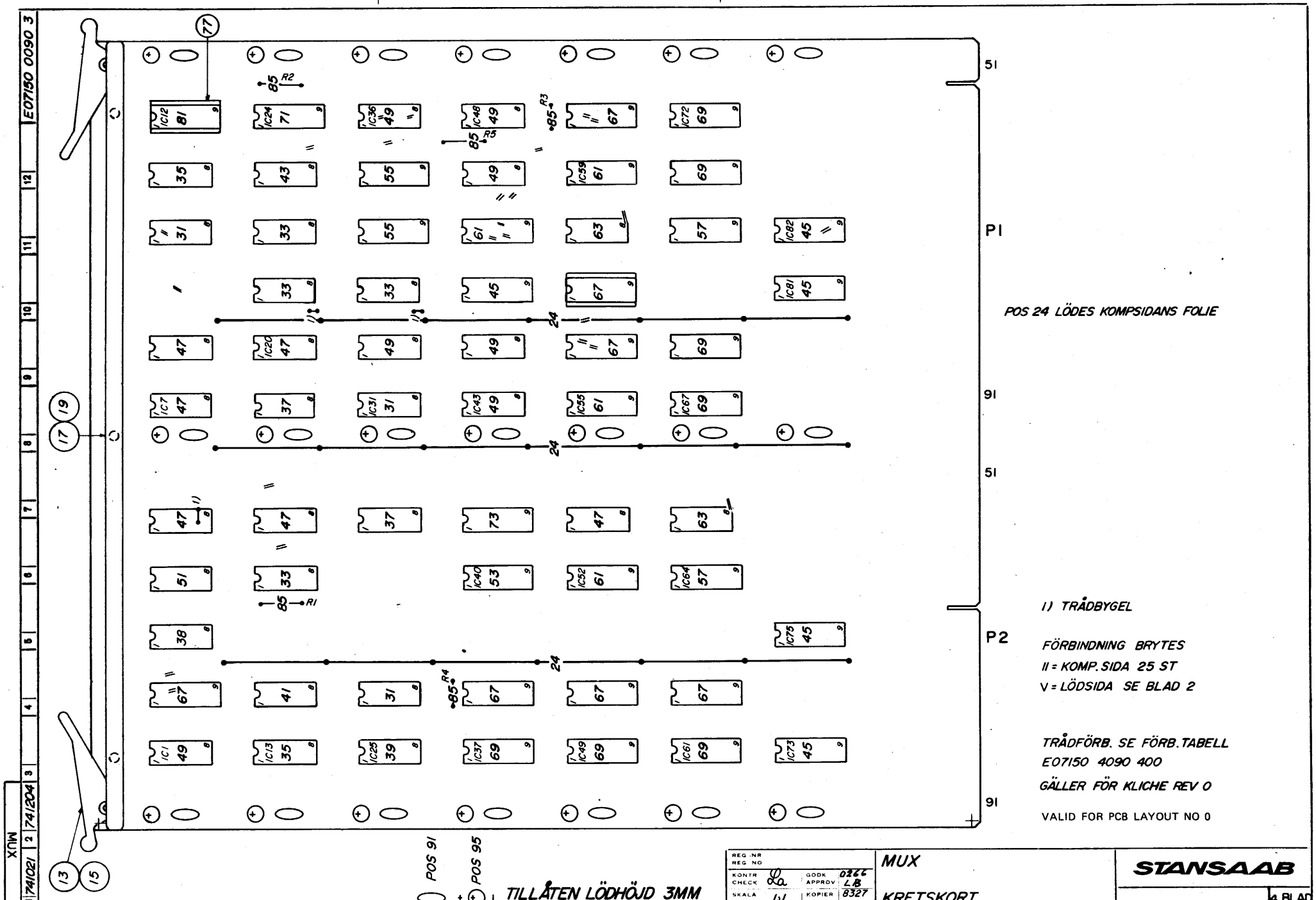
MUX
Multiplexor

STANSAAB
E07150 2090 1
4 BLAD SHEET 3



This document must not be copied, reproduced or disclosed to any third party, either wholly or in part, without our consent. Any infringement will be prosecuted.
STANSAAB
Denna handling får ej utan vårt medgivande, helt eller delvis, kopieras, utlämnas eller användas av obehörig. Överträdelse beivras enligt lag.
STANSAAB
OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!

INGÅR ÄVEN I ALSO USED ON				REG-NR REG. No.	
KONTR CHECK	8422	GODK APPROV	---	MUX Multiplexor	STANSAAB
SKALA SCALE		KOPIER COPIED	8839 8A0		E07150 2090 1
UTFÖRD ISSUED	8427 LA	731105			BLAD SHEET 4



Denna handling får ej utan vårt medgivande bekantgöras kopieras mångfaldigas eller eljest obehörigen utnyttjas. Överträdelse beivras med stöd av gällande lag.
STANSAAB Elektronik AB

OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!

This document must not be copied, reprinted or disclosed to any third party, either wholly or in part, without our consent. Any infringement will lead to legal proceedings.
STANSAAB Elektronik AB SWEDEN

XNW												
REV. 1	741021	2	741204	3	4	5	6	7	8	9	10	11	12	E07150 0090 3

FÖRST ANVÄND I FIRST USED ON	INGÅR ÄVEN I ALSO USED ON		

REG.-NR REG. NO			
KONTR CHECK	Ola	GODK APPROV	0266 LB
SKALA SCALE	1:1	KOPIER COPIED	8327 KGB
UTFÖRD ISSUED	8427 LA		74 04 08

MUX

KRETSKORT
PRINTED BOARD ASSEMBLY

STANSAAB

E07150 0090 3

4 BLAD
1

This document must not be copied, reprinted or disclosed to any third party, either wholly or in part, without our consent. Any infringement will lead to legal proceedings.
STANSAAB Elektronik AB SWEDEN

OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!

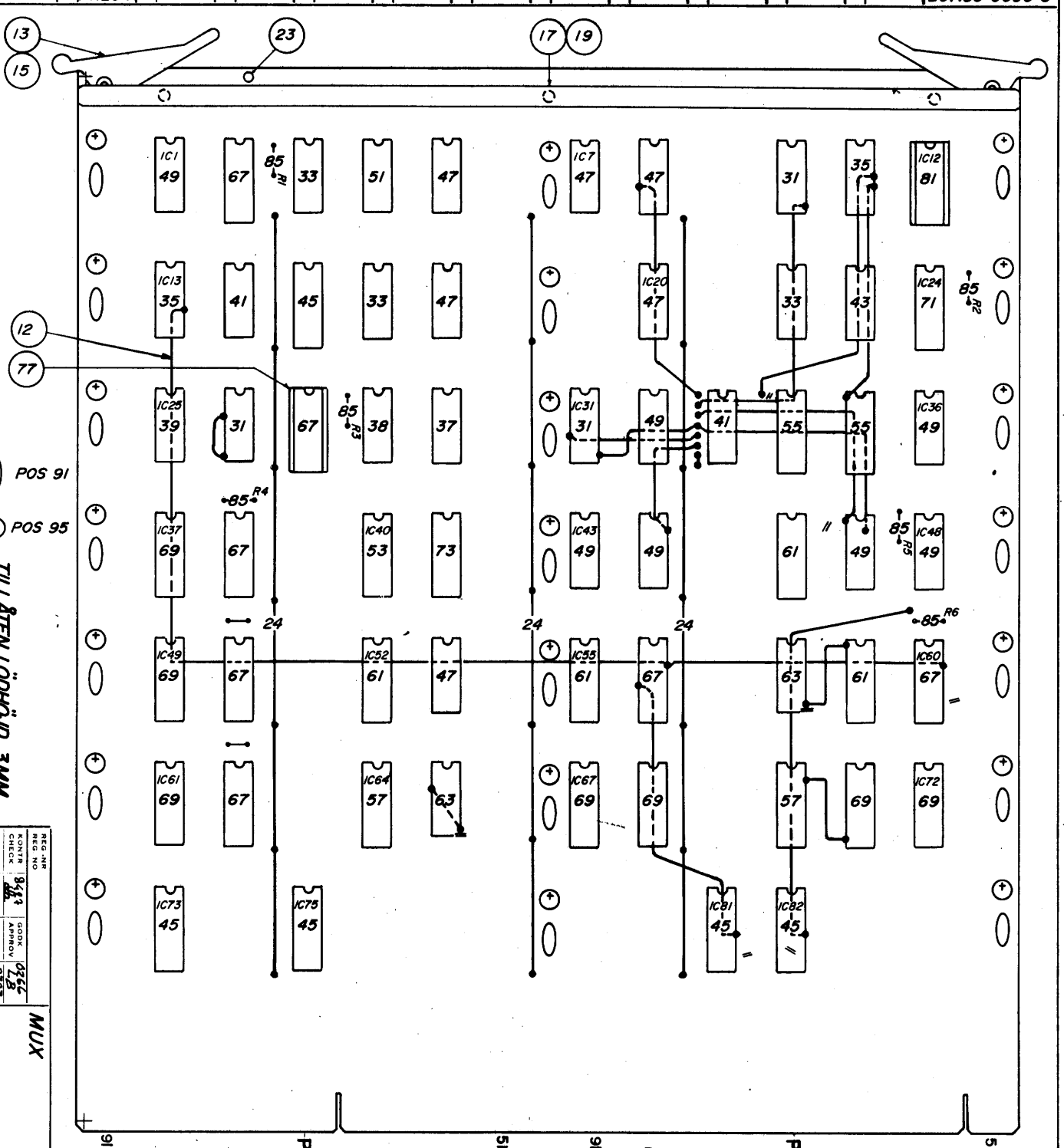
Denna handling får ej utan vårt medgivande bekantgöras kopieras mångfaldigas eller eljest obehörigen utnyttjas. Överträdelse beivras med stöd av gällande lag.
STANSAAB Elektronik AB

MUX

REV. 1 | 741021 | 2 | 741204 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | E07150 0090 3

FÖRST ANVÄND I
FIRST USED ON

INGÅR ÄVEN I
ALSO USED ON



FÖRBINDNINGEN BRYTES
// = KOMP SIDA 7 ST
V = LÖDSIDA SE BLAD 2
GÄLLER FÖR KLICHÉ REV 1

POS 24 LÖDES KOMPSIDANS FOLIE (POS 12) TRÅDFÖRBINDN. 20ST.

VALID FOR PCB LAYOUT NO 1

REG. NR / REG. NO			
KONTR / CHECK	8427 LA	GODK / APPROV	8427 LA
SKALA / SCALE	1:1	KOPIERT / COPIED	8427 KGB
UTFÖRD / DRAWN	8427 LA		
	74 03 21		

MUX
KRETSKORT
PRINTED BOARD ASSEMBLY

STANSAAB

E07150 0090 3

4 BLAD 3

Denna handling får ej utan vart medgivande bekantgoras, kopieras eller på annat sätt mångfaldigas. Överträdelse beivras med stöd av gällande lag.
STANSAAB elektronik ab

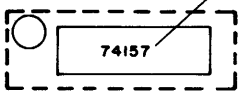
This document must not be copied, reprinted or disclosed to any third party either wholly or in part, without our consent. Any infringement will lead to legal proceedings.
STANSAAB elektronik ab SWEDEN

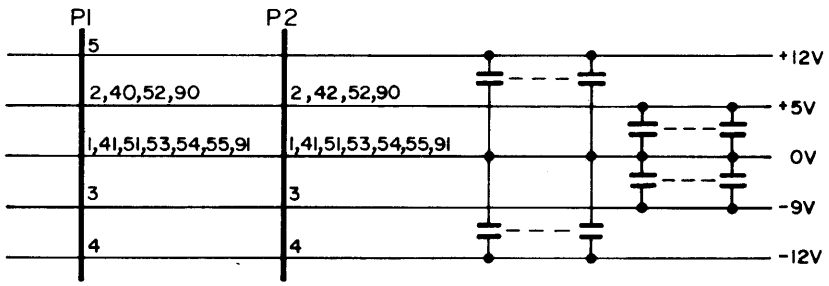
TEST POINTS

NR	SIGNAL
1	
2	
3	
4	BYTE 2
5	14320,080 KHz
6	C9
7	CPC
8	
9	
10	CONDITION SELECT "YES"
11	COUNT SWEEPCOUNTER
12	MAIN ADDRESS REG BIT 0
13	—— " —— 1
14	—— " —— 2
15	—— " —— 3
16	EN POSITION PROGRAM
17	CLC
18	
19	
20	
21	
22	
23	CSC
24	MAIN ADDRESS REG BIT 4
25	EN MAIN PROGRAM
26	V-SYNC
27	H-SYNC
28	BYTE 1
29	
30	
31	
32	
33	
34	
35	
36	
37	
38	
39	
40	
41	



INDICATED NUMBERS ONLY
EXAMPLES OF EXISTING
CIRCUIT TYPES





REVISION OF DIAGRAMS

DIAGRAM AND DRAWING NUMBER	SH	STATUS MOD.	-
E07150 2100 3	1	REV.	
x)		"	
E07150 2100 1	3	"	1
— " — 1	4	"	
		"	
		"	
		"	
		"	
		"	
		"	

x) Block diagram see drawing E 07150 2090 2 sh 2

INGÅR AVEN I / ALSO USED ON			REG.-NR / REG. No
FÖRST ANVÄND I / FIRST USED ON			
KONTR / CHECK Ola	GODK / APPROV CB	TIM PRESENTATION PROGRAM	STANSAAB
SKALA / SCALE	KOPIER / COPIED LD		E 07150 2100 3 — 4 sh
UTFÖRD / ISSUED 8427 LA	731105		1

TIM
REV.I

L1 330 µH · C3 100 nF · R2 100k · Y1 2N2222A · Y2 2N2369 · C1 100pF · C2 18pF · R3 2,2 k · R4 150Ω · X1 14320,80 kHz · R1 39 k · +12V · +5V · 0V

Column counter · 8-BIT 74164 · TTL/2 · TTL/3 · TTL/5

DIM to line store · CP generator · 74157 · JKMS · 74125

50 Hz divider · 7492 · 74193 · GR · GR delay FF · 1Hz · CP line store

Main program output D-time · 10/sh 4 · 5/sh 4

P1	P2		
	89	Write	DA,PA
	38	— " —	MPU
	57	Format line video	DA
86		DA or PA MEMA decoded	MUX
81		16 Hz	MUX
85		Increment MAC	"
	37	Stop CP counter	MPU
	89	F-blink	DA
	8	CP line store	DA
	7	Inhibit video	DA

Pos progr output 11/sh 4

MUX CP/2 35 · DA 40 char/line 66 · " DIM to line store 88 · MUX " 36 · DA CP 16 · " Column 9 17 · MPU General reset 39

3/sh4 Columns · 4/sh4 GR

Position counter · 74193 · Position program counter · 7493 · Main address counter · Main address register · 74174 · Address selector · 74157 · IM 5610 · ROM1 · ROM2 · Yes

Sweep counter · 74193 · Line Counter · CPC · CSC · CLC · CSL · CUL · CUS · LPC · LMA

Sweep counter: 0 bo · " — " — 1 b1 · " — " — 2 b2 · " — " — 3 b3 · P2 9

DA Sweep 9 P2 59 · DA Sweep 4 58

9/sh4 count

Load decoder · 74155 · En main progr · En pos progr · Program activate

ROM 2 out 1/sh 4 · GTP 6/sh 4 · Cond select 8/sh 4 · Program activ 2/sh 4 · Carry CSC 7/sh 4

( ) = Repeated information

Denna handling får ej utan vårt medgivande, bestående, kopieras, reproduceras eller eljest obehörigen användas. Överträdelse beivras enligt lag. STANSAAB äger detta ritningen.

This document must not be copied, reproduced or disclosed to any third party, either wholly or in part, without our consent. Any infringement will be prosecuted. STANSAAB owns the rights to this drawing.

OM NÅGOT ÄR OKLART: FRÅGA! IF IN DOUBT: ASK!

INGÅR ÄVEN I ALSO USED ON				REG.-NR REG. No	
KONTR CHECK	LO	GODK APPROV	0266	STANSAAB	
SKALA SCALE		KOPIER COPIED	0873 LD	TIM PRESENTATION PROGRAM	E07150 2100 1
UTFÖRD ISSUED	8427 LA	731105			4 BLAD SHEET 3

REV. 1 741206 TIM

SCANDIAZO VELVETEX

This document must not be copied, reproduced or disclosed to any third party, wholly or in part, without our consent. Any infringement will be prosecuted under STANDARD element 0 - SWEDEN

OM NÅGOT ÄR OKLART: FRÅGA IF IN DOUBT: ASK!

Denna handling får ej utan vårt medgivande betecknas, kopieras eller lämnas ut till obehörig. Överträdelse beivras enligt STANDARD element 0

INGÅR ÄVEN I ALSO USED ON			73110 5	ISSUED L.A.	4 BLAD SHEET 4
KONTR CHECK	Ulf 9266	GODK APPROV		UTFÖRD DRAWN	E07150 2100 1
SKALA SCALE		KOPIER COPIED			

TIM

PRESENTATION PROGRAM

STANSAAB

REG - NR REG No

REV

TIM

DA Cmd bit timing 71

DA Block video cmp 21

MUX Byte 1+2 31

P1 P2

Last bit generator

Main program output

ELB Position program output

+TTL/5 0V

Position progr output 11/sh3

Main progr output 10/sh3

Byte 2 15

Pos 15 65

Display condition

MPU stop condition

H-sync

DIM to line store

pos 85

pos 100

pos 53

pos 21

pos 15

pos 01

Pos 15

Byte 2

Format line

NO-OP pos 100

CUL

ELB-condition

Byte 0

Byte 1

V-sync pos 01

V-sync

Last sweep

D-time 5/sh3

Displ condition 62

Format line 63

Pos 01 V-sync 22

Byte 0 13

Byte 1 14

Last displ sweep 60

Displ condition 61

Pos 21 64

V+H-sync 67

Byte 1+2 18

Clear I/O-reg 19

Load I/O-reg 20

I/O shiftpulse 68

Load data reg 70

Load status reg 69

GR 4/sh3

Columns 3/sh3

Program out 2/sh3

ROM 2 out 1/sh3

Count 9/sh3

Cond sel 8/sh3

CARRY 7/sh3

GTP 6/sh3

ELB-time

pos 85

Group matrix

Subject matrix

Condition selector

This document must not be copied, reprinted or disclosed to any third party, either wholly or in part, without our consent.
Any infringement will lead to legal proceedings.
STANSAAB Elektronik AB SWEDEN

OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!

Denna handling får ej utan vårt medgivande beträffande kopiering mångfaldigas eller alstet obehörigen utnyttjas. Överträdelse bestraffas med stöd av gällande lag.
STANSAAB Elektronik AB

Denna handling får ej, utan vårt medgivande bekantgöras kopieras mångfaldigas eller eljest obehörigen utnyttjas. Överträdelse beivras med stöd av gällande lag.
STANSAAB Elektronik AB

OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!

This document must not be copied, reprinted or disclosed to any third party, either wholly or in part, without our consent. Any infringement will lead to legal proceedings.
STANSAAB Elektronik AB SWEDEN

TIM

REV.			
1	74 10 16		
2	74 12 06		

E07150 0100 3

TRÅDFÖRBINDNING LÖDSIDA
(SE BLAD 2)

POS 24 LÖDES KOMPSIDANS FOLIE
(POS 12) TRÅDFÖRBINDN. 15 ST

FÖRBINDNINGEN BRYTES
II = KOMPSIDA 7 ST.
Λ = LÖDSIDA (SE BLAD 2).

POS 127, 157 FIXERAS MED LIM
ENL. F4714
TILLÅTEN LÖDHÖJD 3MM

GÄLLER FÖR KLICHE REV 1.
VALID FOR PCB LAYOUT NO 1.

POS 113

POS 119

FÖRST ANVÄND I
FIRST USED ON

INGÅR ÄVEN I
ALSO USED ON

REG.-NR / REG NO			
KONTR / CHECK		GODK / APPROV	8266 LB
SKALA / SCALE	1:1	KOPIER / COPIED	8327 KGB
UTFÖRD / ISSUED	8427 LA	74 03 25	

TIM

KRETSKORT
PRINTED BOARD ASSEMBLY

STANSAAB

E07150 0100 3

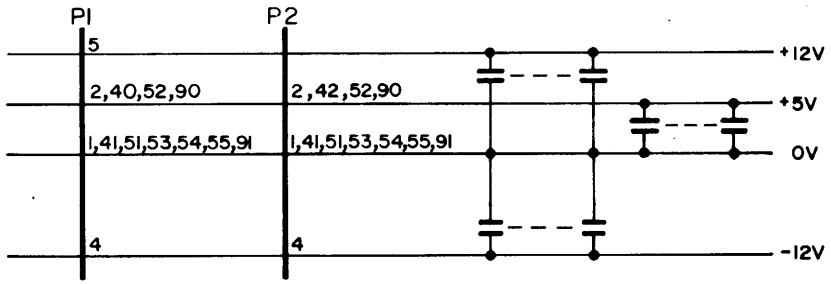
4 BLAD 3

Denna handling får ej utan vårt medgivande bekantgöras, kopieras, mangfaldigas eller eljest obehörigen utnyttjas. Överträdelse beivras med stöd av gällande lag
STANSAAB elektronik ab

TEST POINTS	
NR	SIGNAL
1	Nul message
2	Byte Ö
3	Erable DA
4	Load video register
5	Underline/Lightpen detectable line
6	Interrupt
7	
8	
9	
10	
11	
12	High intensity
13	Inhibit video
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	
32	
33	C9
34	CP line store
35	Non displayed field
36	CP
37	Underline/Blink
38	Lightpen detectable field
39	-3,3 V
40	Write DIM
41	Serial data



INDICATED NUMBERS ONLY
EXAMPLES OF EXISTING
CIRCUIT TYPES



REVISION OF DIAGRAMS AND ASSEMBLY DRAWING

DIAGRAM AND DRAWING		STATUS	
NUMBER	SH.	MOD.	
E07151 2110 3	1	REV.	
—"— 3	2	"	
—"— 1	3	"	
—"— 1	4	"	
—"— 7	5	"	
		"	
		"	
		"	
		"	
		"	

This document must not be copied, reprinted or disclosed to any third party either wholly or in part without our consent. Any infringement will lead to legal proceedings.
STANSAAB elektronik ab SWEDEN

INGÅR ÄVEN I / ALSO USED ON			REG.-NR / REG. No
FÖRST ANVÄND I / FIRST USED ON			
KONTR / CHECK	AG	GODK / APPROV	0220
SKALA / SCALE		KOPIER / COPIED	8839 BA 0
UTFÖRD / ISSUED	8489 AG	731011	

DA
Display Adapter

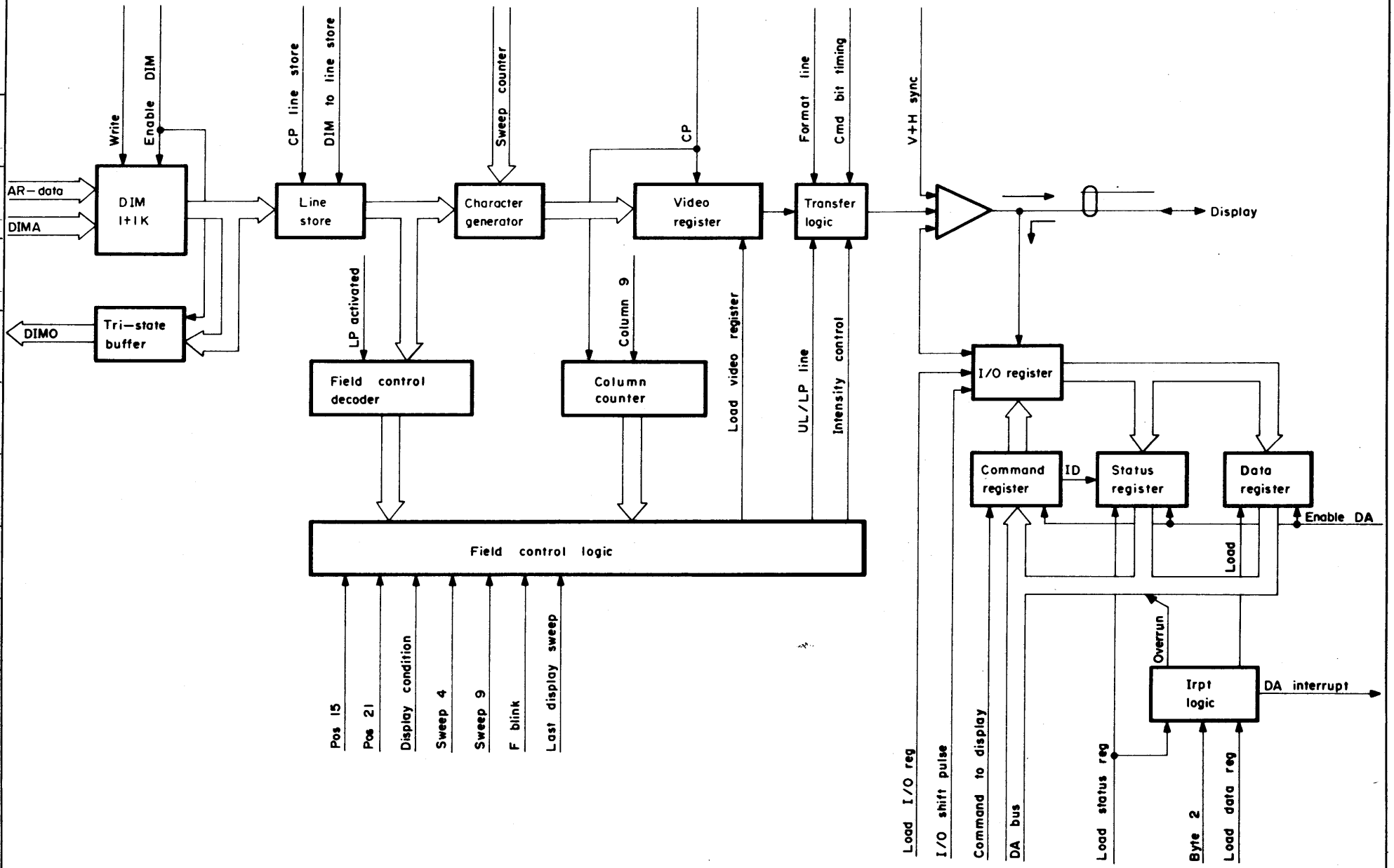
STANSAAB
E07150 2110 3
5 sh
1

Denna handling får ej utan vårt medgivande bekantgöras kopieras mångfaldigas eller eljest obehörigen utnyttjas. Överträdelse beivras med stöd av gällande lag.
STANSAAB Elektronik AB

OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!

This document must not be copied, reprinted or disclosed to any third party, either wholly or in part, without our consent. Any infringement will lead to legal proceedings.
STANSAAB Elektronik AB SWEDEN

DA



INGÅR ÄVEN I ALSO USED ON				REG.-NR REG. No	
KONTR CHECK		GODK APPROV	0266 LB	DA Display Adapter	STANSAAB
SKALA SCALE		KOPIER COPIED	8506 G.S.		E 07150 2110 3
UTFÖRD ISSUED	8628 GE	74 12 02			5 BLAD SHEET 2

REV.1

FÖRST ANVÄND I FIRST USED ON

INGÅR ÄVEN I ALSO USED ON

OMJAM BARRÅ TERRA BL. 301:33

This document must not be copied, reproduced or disclosed to any unauthorized person in any form without our consent.
Any infringement will lead to legal prosecution.
Alla rättigheter förbehålles STANSAAB

OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT - ASK!

Denna handling får ej utan vårt medgivande kopieras, lämnas ut eller användas på otillbörligt sätt.
Sådana överträdelser beivras enligt gällande lag.

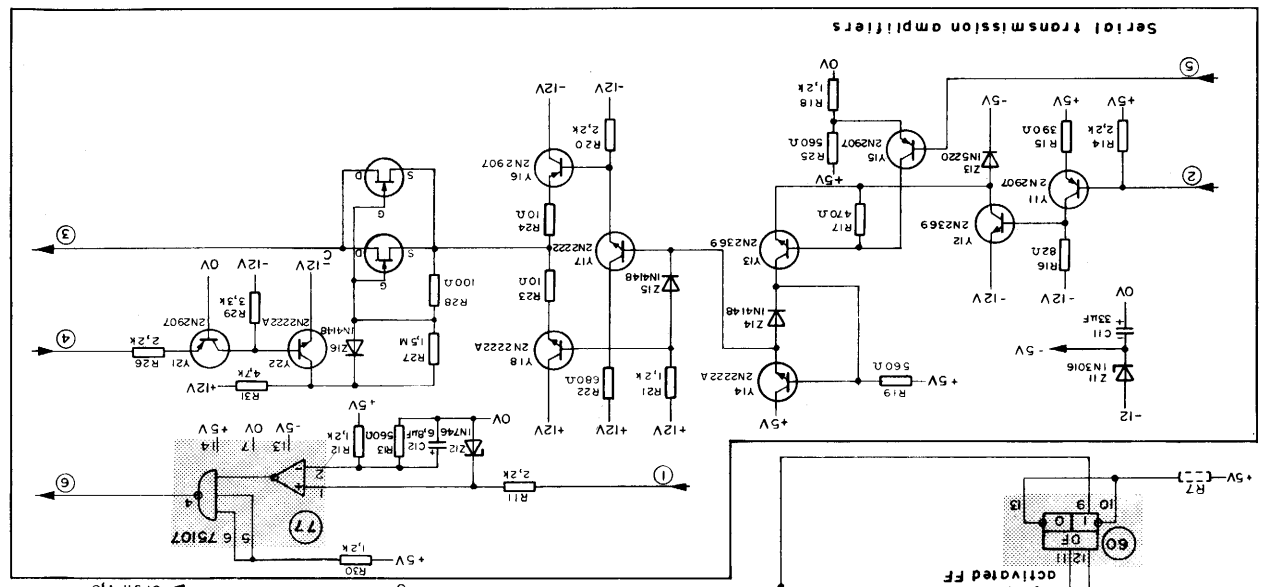
INGÅR EJ I ALSO USED ON					
KONTR CHECK	AS	500K Japprov	8.8.92 KOPIER COPIED	8.8.92 BAO	
SKALA SCALE					
UTGÅR ISSUED	8.8.92 AS	73 10 11			

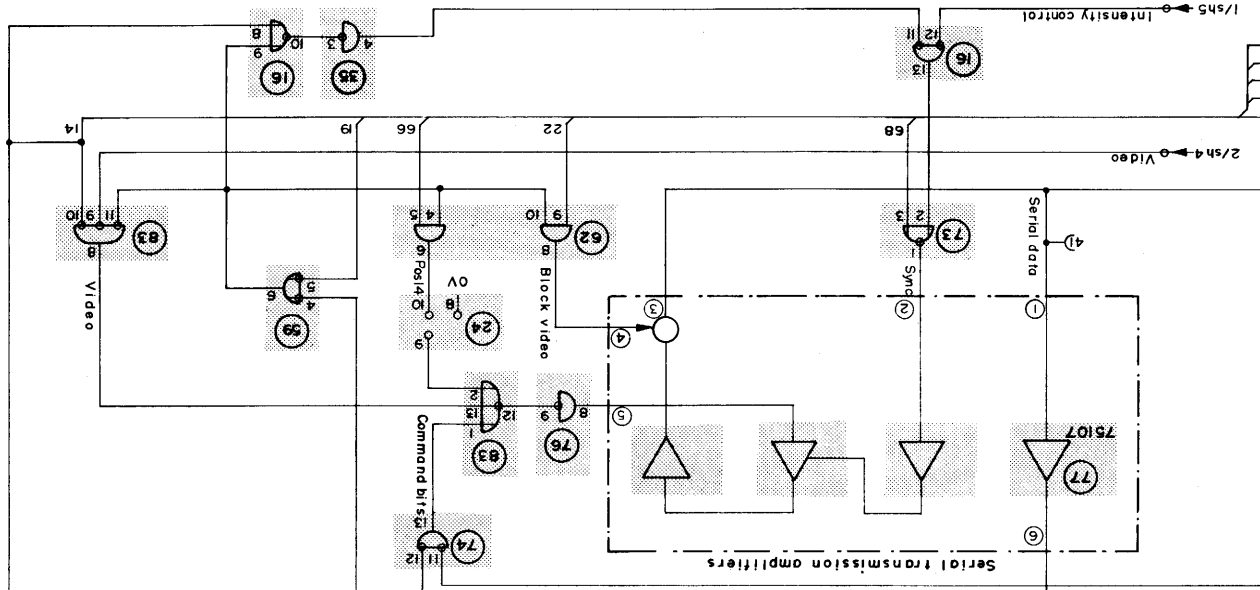
DA
Display Adapter

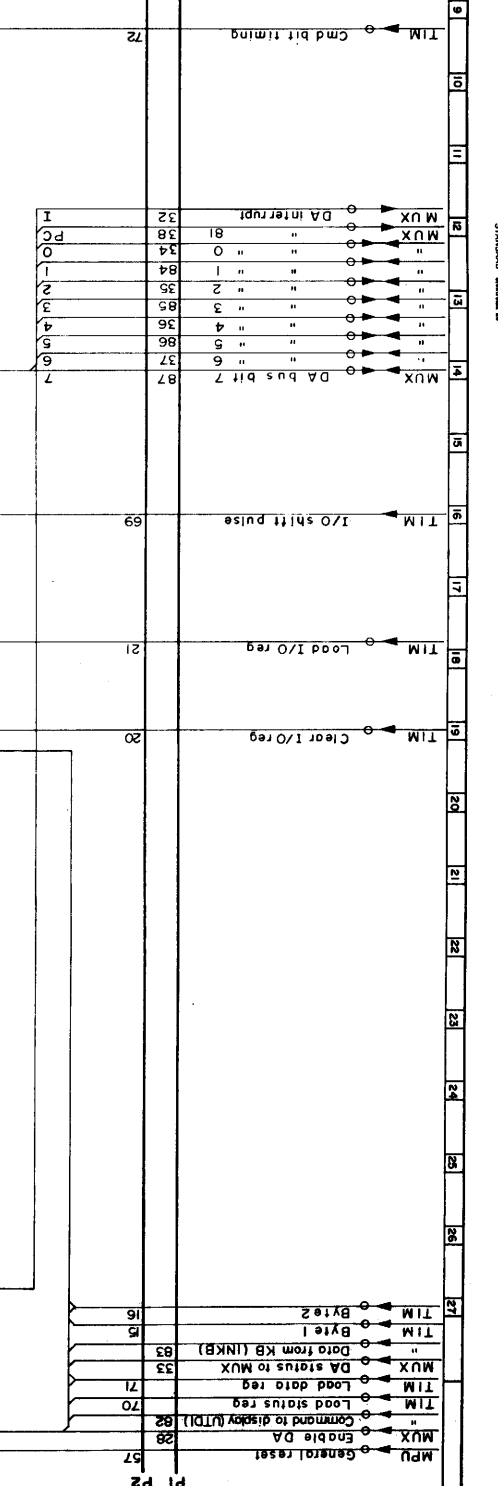
STANSAAB

E07150 2110 1
5 BLAD SHEET 3

REG - NR
REG. No.







Line store

Display memory

Field control decoder

Character generator

Video register

Underline inhibit control

Column counter

Trailing edge delay

P1 P2

TIM C.P line store 9
TIM DIM to line store 39
MPU AR data:0 73
" " 1 23
" " 2 72
" " 3 22
" " 4 71
" " 5 21
" " 6 70
" " 7 20
TIM Write 88
MUX Enable DIM 29
MUX DIMA IO 38
" " 9 87
" " 8 37
" " 7 86
" " 6 36
" " 5 85
" " 4 35
" " 3 84
" " 2 34
" " 1 83
" " 0 33
MEMI DIMO Bit 0 77
" " " 1 27
" " " 2 76
" " " 3 26
" " " 4 75
" " " 5 25
" " " 6 74
" " " 7 24
TIM CP 17
TIM Column 9 18
TIM Sweep counter: 0 10
" " 1 11
" " 2 12
" " 3 13
TIM Format line video 58
3/sh3 Lightpen activated
GRALFA Video 89
TIM Inhibit video 8

Write DIM

RWM 1024x1 BIT 2102

IC 50 IC 51 IC 52 IC 53 IC 54 IC 55 IC 56

IC 63, 64, 65, 66, 67, 68, 69, 70

2 nd k-Ram

1:st k-Ram

IC 49

+5V 0V

0-7

TMS 3409

4x80 BIT

MCA DAO MCB DBO MCC DCO MCD DDO

XA79 XB79 XB79 XD79

MCM 6570L/5CM 10093

-12V

R32 330Ω 0,5W

-3,3V

Z17 IN746A

0V

DA-ROM1

DA-ROM2

7442

1-OF-10

ROM 58x8 BIT IM5610

B7 B7B6x10 LP detectable field Non displayed field High intensity Normal intensity

Space SF Cursor [ ] M Nul

Field control characters 7 sh/5

Video register

Load video register 6/sh5

8-BIT 74165

Underline inhibit control

4-BIT 74175

R6 +5V

Column counter

8-BIT 74164

R4 +5V

Colums 8/sh5

Trailing edge delay

R3 +5V

R9 1,2k +5V

Format line 9/sh5

UL/LP line 10/sh5

Video 2/sh3

Inhibit video 11/sh5

OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!

This document must not be copied, reprinted or disclosed to others without written permission from SAAB-SCANIA. Any infringement will lead to legal prosecution.

INGÅR ÄVEN I ALSO USED ON				REG-NR REG. No	
KONTR CHECK	AG	GODK APPROV		STANSAAB	
SKALA SCALE		KOPIER COPIED	BB13 BA0	DA Display adapter	EQ7150 2110 1
UTFÖRD ISSUED	B489 AG	73 10 11			5 BLAD SHEET 4

7/sh4 Field control characters
8/sh4 Columns
C2 C7 C6
H C D K A H G N L M B E F C9 C7
5F
Non displayed field
Cursor
B7 5F
Space
Nul
[]
B7B6=10
High intensity
Normal intensity
S4 Lightpen detectable field
DOG 4-BIT 3/4-74175 Q0 Q̄ Q2 Q̄
+5V R1
P1 P2
TIM Pos 21 65
Block 5F
Intensity control logic
Intensity FF
JKMS
Underline blink FF
Lightpen detectable field FF
Non displ. field FF
+5V R2
0V
Underline/ Blink stop
TIM Format line 64
TIM Displ condition 62
TIM Displ condition 63
D D̄
3/sh3 Lightpen activated
4-BIT 1/2-74175
4-BIT 1/4-74175
R4 +5V
Field termination
Inhibit video 11/sh4
Format line 9/sh4
P2
61 Last displ sweep TIM
66 Pos 15 TIM
67 40 characters/line TIM
Load video register 6/sh4
C1 C7 C2 C3 C5 C9 CP
TIM Sweep 4 59 S4
TIM Sweep 9 60 S9
Underline/ blink
TIM F blink 89
R3 +5V
UL/LP Line FF
intensity control 1/sh3
UL/LP line 10/sh4
OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!
Denna handling får ej utan vårt medgivande kopieras, kopieras mångfaldigas eller på annat obehörigen användas. Överträdelse beivras av STANSAAB Elektronik AB
This document must not be copied, reproduced or disclosed to any third party, either wholly or in part, without our consent. Infringement will be prosecuted by STANSAAB Elektronik AB SWEDEN
REV. DA
INGÅR ÄVEN I
ALSO USED ON
FÖRST ANVÄND I
FIRST USED ON
KONTR CHECK AG TOOLS APPROV
SKALA SCALE
KOPIER COPIED 8839 BAd
UTFÖRD ISSUED 8489 AG 73 10 11
REG.-NR REG. No
D A
Display adapter
STANSAAB
E07150 2110 7
5 BLAD SHEET 5

This document must not be copied, reprinted or disclosed to any third party, either wholly or in part, without our consent. Any infringement will lead to legal proceeding.
STANSAAB Elektronik AB SWEDEN

DA

OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!

Denna handling får ej utan vårt medgivande beträffande kopiorna mångfaldiga eller alleat obehörigen utnyttjas. Överträdelse beivras med stöd av gällande lag.
STANSAAB Elektronik AB

REV. 1 741202 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | E07150 0110 3

FÖRST ANVÄND I / FIRST USED ON

INGÅR ÄVEN I / ALSO USED ON

STANSAAB

E07150 0110 3

4 BLAD | 1

DA

KRETSKORT
PRINTED BOARD ASSEMBLY

REG.-NR / REG. NO			
KONTR / CHECK		GODK / APPROV	0226 LB
SKALA / SCALE	1:1	KOPIER / COPIED	8327 KGB
UTFÖRD / ISSUED	8902 SÄF		74 05 27

GÄLLER FÖR KLICHE REV 0

VALID FOR PCB LAYOUT NO 0

POS 101 ISOLERAS MED SKYDDSREMSA
ENL. POS 29

TRÅDFÖRB. SE FÖRB. TABELL
E07150 4110 400

FÖRBINDNING BRYTES
II = KOMP. SIDA 26 ST
V = LÖDSIDA SE BLAD 2

I) = TRÅDBYGEL 9 ST

TILLÄTEN LÖDHÖJD 3MM

POS 141

POS 143

P1

P2

This document must not be copied, reprinted or disclosed to any third party, either wholly or in part, without our consent. Any infringement will lead to legal proceedings.
STANSAAB Elektronik AB SWEDEN

OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!

Denna handling får ej utan vårt medgivande bekantgöras kopieras mångfaldigas eller eljest obehörigen utnyttjas. Överträdelse beivras med stöd av gällande lag.
STANSAAB Elektronik AB

REV. 1 | 741202 | E07150 0110 3

FÖRST ANVÄND I / FIRST USED ON

INGÅR ÄVEN I / ALSO USED ON

POS 141

POS 143

TILLÅTEN LÖDHÖJD 3MM

DA

KRETSKORT
PRINTED BOARD ASSEMBLY

E07150 0110 3

STANSAAB

4 BLAD 3

REG.-NR / REG. NO

KONTR / CHECK

SKALA / SCALE 1:1

UTFÖRD / ISSUED 8902 SAF

74 08 14

GODK / APPROV 02 6 C LS

COPIER 8327 KGB

KOMPONENTBEN Z12 VIKES

POS 101 ISOLERAS MED SKYDDSREMSA ENL. POS 29

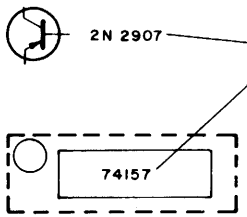
VALID FOR PCB LAYOUT NO 1

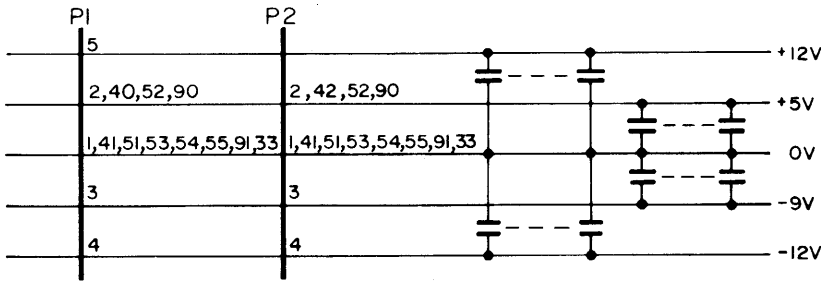
GÄLLER FÖR KLICHÉ REV 1

HÅL RENSBORRAS

TEST POINTS

NR	SIGNAL
1	$\overline{\text{Set status}}$
2	Reg to send to PR
3	Count down CP
4	Data received
5	Interrupt
6	Printer data
7	Load transmitter
8	Reg to send from PR
9	Data to PR (UTPR)
10	Transmitter rate
11	Buffer empty
12	CEO
13	$\overline{\text{PC-error}}$
14	Set PC-error
15	CEI
16	Print out
17	Enable PA
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	
32	
33	
34	
35	
36	
37	
38	
39	
40	
41	





REVISION OF DIAGRAMS

DIAGRAM AND DRAWING NUMBER	SH.	STATUS MOD.
E07150 2120 3	1	REV.
" 3	2	"
" 7	3a	"
" 7	3b	"
		"
		"
		"
		"
		"
		"

PA

REVI

INGÅR AVEN ALSO USED ON		REG. NO
FÖRST ANVÄND FIRST USED ON		
KONTR CHECK	9496 Hdl.	GODK APPROV. 0266 LB
SKALA SCALE		KOPIER COPIED 9839 BA6
UTFÄRD. ISSUED	9496	KET 73 10 31

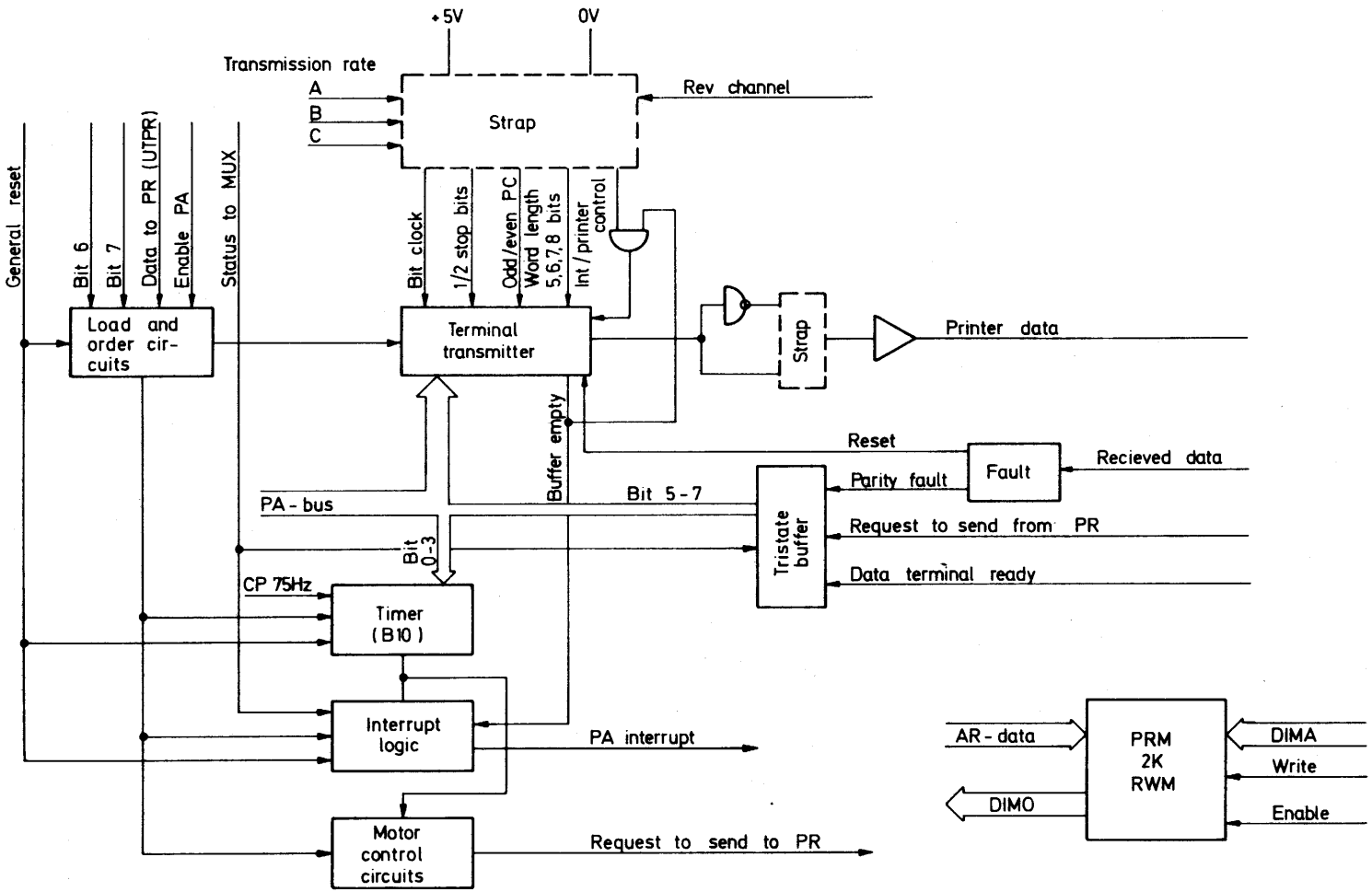
PA
Printer adapter

STANSAAB

E07150 2120 3 | 3sh | 1

Denna handling får ej utan vårt medgivande bekantgöras kopieras mångfaldigas eller eljest obehörigen utnyttjas. Överträdelse beivras med stöd av gällande lag.
STANSAAB Elektronik AB

OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!



This document must not be copied, reprinted or disclosed to any third party, either wholly or in part, without our consent. Any infringement will lead to legal proceedings.
STANSAAB Elektronik AB SWEDEN

INGÅR ÄVEN I ALSO USED ON				REG.-NR REG. No	
KONTR CHECK	7496 Kal	GODK APPROV	0266 CB	PA Printer Adapter	STANSAAB
SKALA SCALE		KOPIER COPIED	ACL 8653		E 07150 2120 3
UTFÖRD ISSUED	9496 KET	74 11 05			3 BLAD SHEET 2

FÖRST ANVÄND I FIRST USED ON | INGÅR ÄVEN I ALSO USED ON | REV.1

This document must not be copied, reproduced or disclosed to any third party, either wholly or in part, without our consent.
STANSAAB Elektronik AB SWEDEN

OM NÅGOT ÄR OKLART, FRÅGA!
IF IN DOUBT, ASK!

Denna handling får ej utan vårt medgivande mångfaldigas, kopieras mekaniskt eller på annat sätt användningen utnyttjas. Överträdelse medför skadeståndsskyldighet.
STANSAAB Elektronik AB

SCHEMATA WIRING

REV.		
	MUX	PA status to MUX
	PR	Data terminal ready
	PR	Req. to send from PR
	PR	Rev. channel
	MUX	75 Hz
	MPU	General reset
	"	Enable PA
	MUX	Data to PR (UPR)
	"	" 7
	"	" 6
	"	" 5
	"	" 4
	"	" 3
	"	" 2
	"	" 1
	MEM	DI MO 0
	"	" b9
	"	" b8
	"	" b7
	"	" b6
	"	" b5
	"	" b4
	"	" b3
	"	" b2
	"	" b1
	"	" b0
	MUX	Dimp b10
	TIM	Write
	MUX	PA memory enable
	"	" 7
	"	" 6
	"	" 5
	"	" 4
	"	" 3
	"	" 2
	"	" 1
	MPU	AR-data 0
	"	" 7
	"	" 6
	"	" 5
	"	" 4
	"	" 3
	"	" 2
	"	" 1
	"	PA Bus 0
	"	" 7
	"	" 6
	"	" 5
	"	" 4
	"	" 3
	"	" 2
	"	" 1
	"	" A
	MUX	Transmitter role B
	"	" C
		P1

KONTR CHECK	4992	600K APPROV	0266 Z8	STANSAAB
UTFÖRD	9496	KOPIEN COPIED	8839 BAO	PA PRINTER ADAPTER
SKALA SCALE				
ISSUED	K E T	73 10 31		E07150 21207
				3 SHEET 3 BLAD 3d

Voild for PCB layout 1 Mod 1

This document must not be copied, reprinted or disclosed to any unauthorized person or company without written consent.
STANSAAB ELEKTRONIK AB STOCKHOLM

OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!

Denna handling får ej utan vårt medgivande tillhandahållas annan eller användas för annat ändamål än det varför den är avsedd.
STANSAAB ELEKTRONIK AB

KONTR CHECK	GUNNE AKE	9004 9206	PA		
SKALA SCALE	NORLEN COMPILE	88/59 BAO	PRINTER ADAPTER		
UTFÖRD ISSUED	9496 K E T	73 10 31			

STANSAAB

E07150 2120 7 — 3 BLAD SHEET — 3 b

REV.

Valid for PCB layout 0 Mod 0

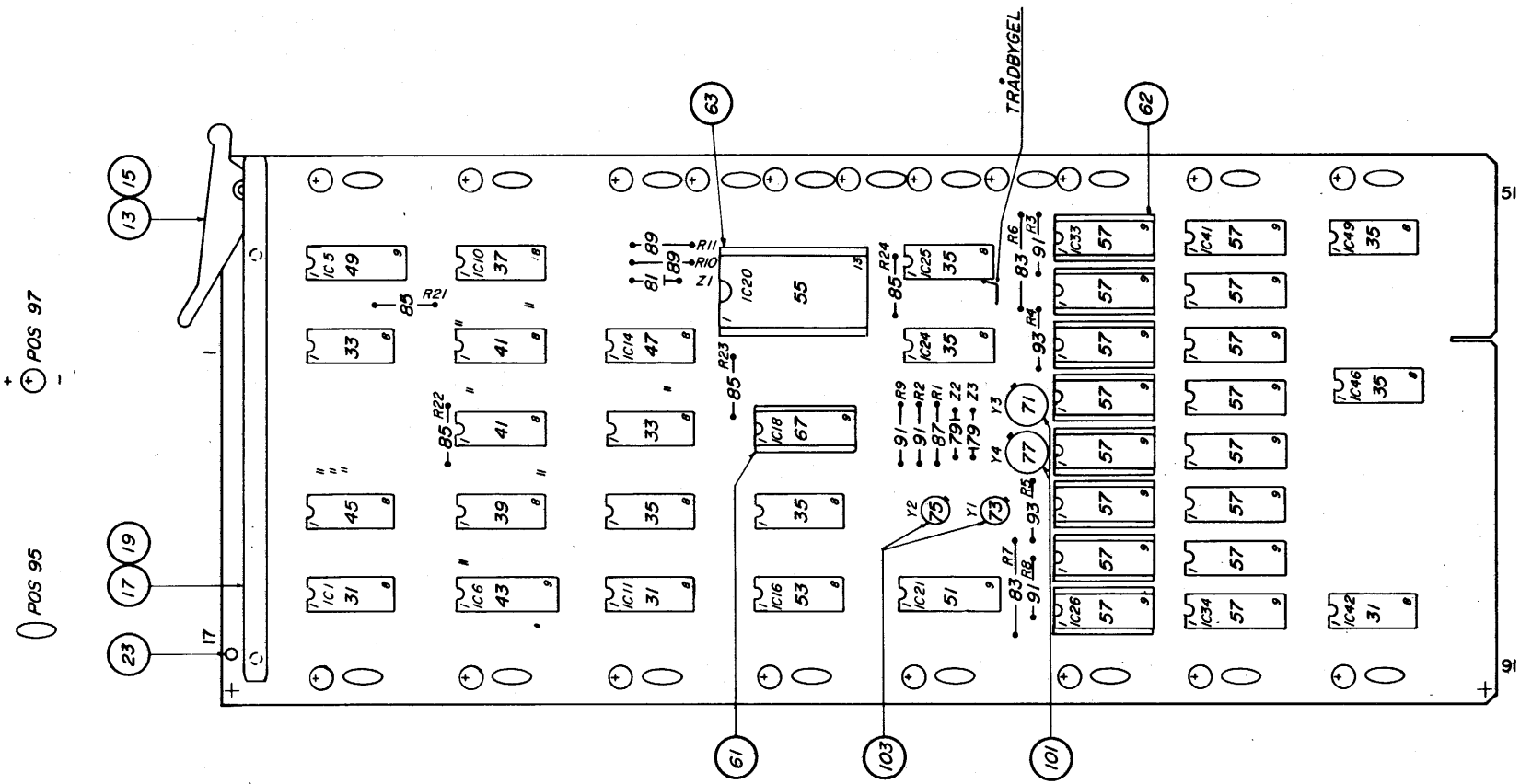
P1	11	PA interrupt	MUX
	37	Received data	PR
	87	Bit clock	
	34	Request to send to PR	PR
	36	Printer data	PR
	61	PA status to MUX	MUX
	38	Data terminal ready	PR
	35	Reg. to send from PR	PR
	39	Rev channel	PR
	66	75Hz	MUX
	89	General reset	MPU
	13	Enable PA	"
	12	Data to PR (UTPR)	MUX
	21	" " 7	"
	71	" " 6	"
	22	" " 5	"
	72	" " 4	"
	23	" " 3	"
	24	" " 2	"
	74	" " 1	MEM
	81	" " b9	"
	31	" " b8	"
	80	" " b7	"
	30	" " b6	"
	79	" " b5	"
	29	" " b4	"
	78	" " b3	"
	28	" " b2	"
	77	" " b1	"
	27	" " b0	"
	32	Dimg bIO	MUX
	82	Write	TIM
	25	PA memory enable	MUX
	17	" " 7	"
	67	" " 6	"
	18	" " 5	"
	68	" " 4	"
	19	" " 3	"
	69	" " 2	"
	20	" " 1	"
	70	AR-data 0	MPU
	60	" " 7	"
	10	" " 6	"
	59	" " 5	"
	9	" " 4	"
	58	" " 3	"
	8	" " 2	"
	57	" " 1	"
	7	PA Bus 0	"
	16	" " A	"
	65	" " C	"
	15	Transmitter rate B	MUX
P1	63		
	16		

Interrupt logic
PC-error
Data received
Timer
Print out FF
RQS-FF
Transmitter
MC 2257
2N2907
2N2222A
2N3613
1N4148
IC 26
IC 27
IC 28
IC 29
IC 30
IC 31
IC 32
IC 33
IC 34-41
74193
7490
+5V
+12V
-12V
0V

Denna handling får ej utan vårt medgivande bekantgöras kopieras helt eller delvis eller på annat sätt obehörigen utnyttjas. Överträdelse beivras med stöd av gällande lag.
STANSAAB Elektronik AB

OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!

This document must not be copied, reprinted or disclosed to anybody in any manner wholly or in part without our consent. Any infringement will lead to legal proceedings.
STANSAAB Elektronik AB SWEDEN



FÖRBINDNING BRYTES

II = KOMP.SIDA 9 ST

V = LÖDSIDA SE BLAD 2

TRÅDFÖRB. SE FÖRB.TABELL

E07150 4120 400

VALID FOR PCB LAYOUT NO 0

GÄLLER FÖR KLICHE REV 0

TILLÅTEN LÖDHÖJD 3MM

REG.NR / REG NO			
KONTR / CHECK	Kef	GODK / APPROV	0266 LB
SKALA / SCALE	1:1	KOPIER / COPIED	8327 KGB
UTFÖRD / ISSUED	9496 KET		74 04 08

PA

KRETSKORT
PRINTED BOARD ASSEMBLY

STANSAAB

E07150 0120 3

4 BLAD 1

REV.														
1	740822	2	741106	3		4		5		6		7		8

9		10		11		12	E07150 0120 3							

FÖRST ANVAND I / FIRST USED ON

INGÅR ÄVEN I / ALSO USED ON

PA

V

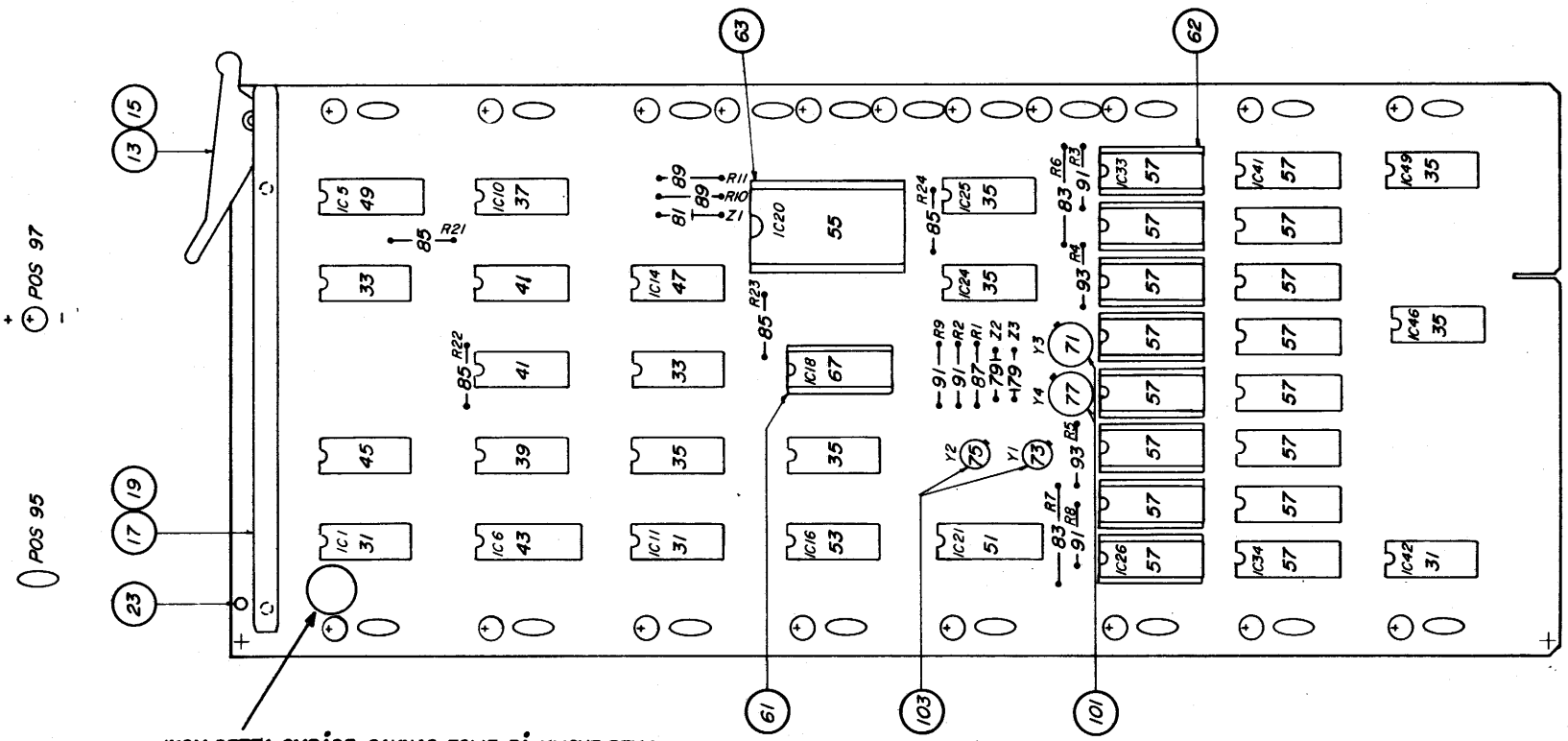
A

Denna handling får ej utan vårt medgivande bekantgöras
kopieras mångfaldigas eller överlåtas i original eller ändrad form
trädelse bevakas med stöd av gällande lag.
STANSAAB Elektronik AB

OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!

This document must not be copied reprinted or disclosed to
any third party, either wholly or in part without our consent.
Any infringement will lead to legal proceedings.
STANSAAB Elektronik AB SWEDEN

E07150 0120 3



INOM DETTA OMRÅDE SAKNAS FOLIE PÅ KLICHE REV I
(REVIDERINGSNUMMER SAKNAS I FOLIET)

VALID FOR PCB LAYOUT NO 1

GÄLLER FÖR KLICHE REV I

TILLÅTEN LÖDHÖJD 3MM

REG-NR / REG NO			PA
KONTR / CHECK		GODK / APPROV	
SKALA / SCALE	1:1	KOPIER / COPIED	8327 KGB
UTFÖRD / ISSUED	9496 KET		741106

KRETSKORT
PRINTED BOARD ASSEMBLY

STANSAAB

E07150 0120 3

4 BLAD 3

REV. 1

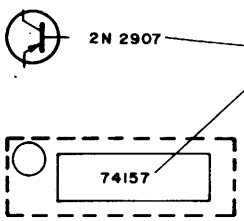
FÖRST ANVAND I / FIRST USED ON

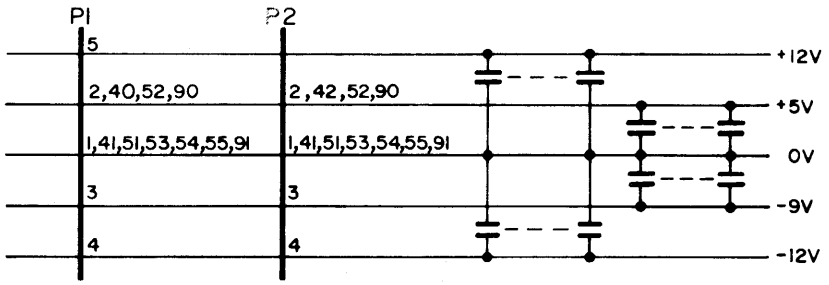
INGÅR ÄVEN I / ALSO USED ON

Denna handling får ej utan vårt medgivande bokantgöras, kopieras, mångfaldigas i vilken form eller del som helst, överlämnas till utomstående eller utnyttjas på otillbörligt sätt. Överträdelse beivras med stöd av gällande lag.
STANSAAB elektronik ab

TEST POINTS

NR	SIGNAL
1	Inhibit clock interrupt
2	0V
3	0V
4	Inhibit time out
5	12,288 MHz
6	
7	Carry
8	Zero
9	OP:1
10	
11	
12	
13	
14	
15	
16	
17	OP:3
18	Mia
19	MEMO:2
20	
21	
22	C3/Pin 11
23	Mij
24	OP:0
25	
26	
27	
28	
29	
30	
31	
32	OP:2
33	Mir
34	End
35	Mis
36	Mit
37	Mie
38	
39	Programrunning level 1
40	—— " —— 0
41	Condition false





REVISION OF DIAGRAMS AND ASSEMBLY DRAWING

DIAGRAM AND DRAWING NUMBER	SH.	STATUS MOD.	5
E 07150 2000 3	1	REV.	
—— " —— 2	2	"	
—— " —— 1	3	"	
—— " —— 1	4	"	
		"	
		"	
		"	
		"	
		"	
E 07150 0001 3		"	4

INGÅR AVEN I / ALSO USED ON

FÖRST ANVÄND I / FIRST USED ON

KONTR CHECK	SB	GODK APPROV	0226 LB
SKALA SCALE		KOPIER COPIED	0673 L-D
UTFÖRD ISSUED	9396 SB		730409

REG.-NR / REG. No

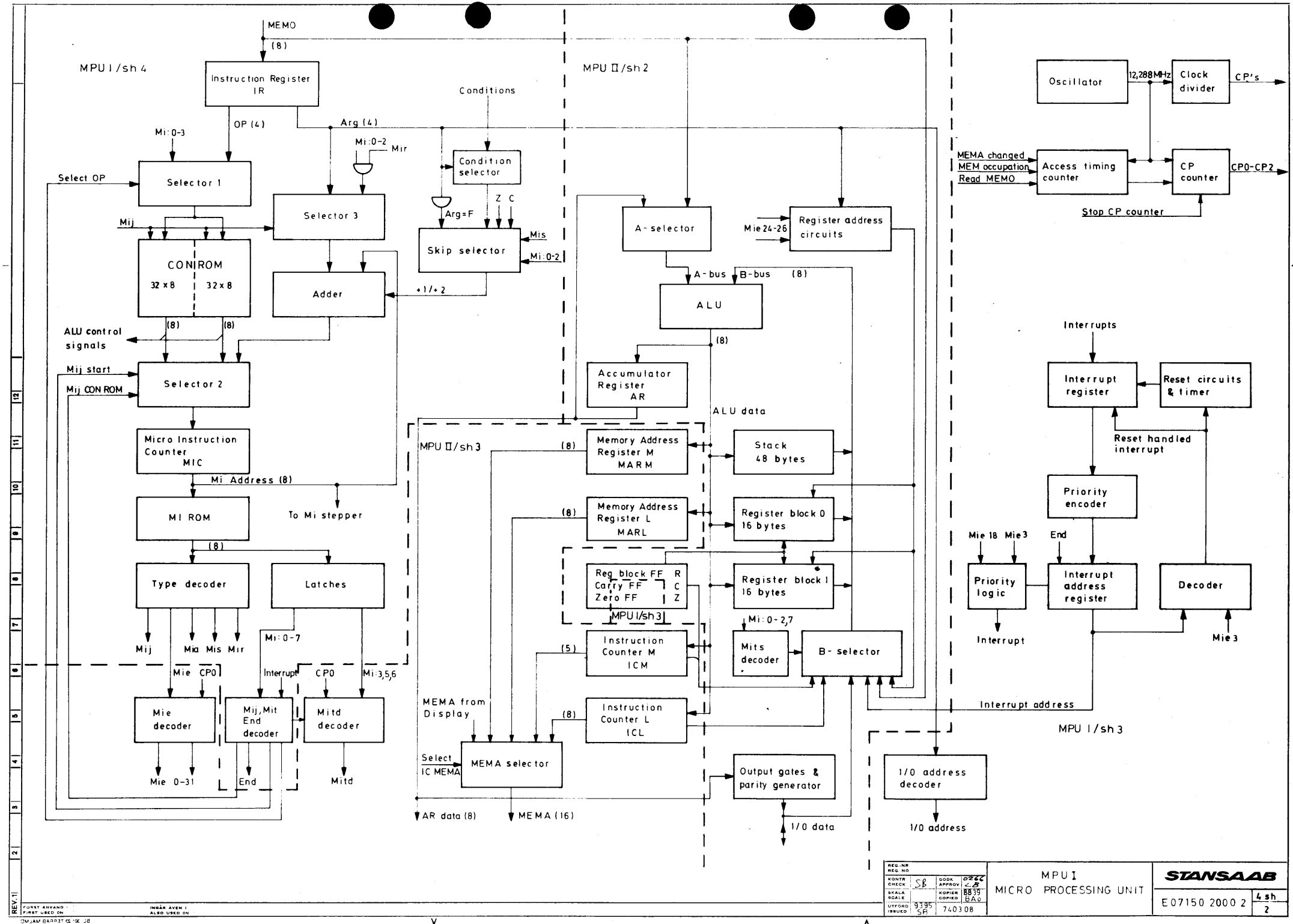
MPU I
MICRO PROCESSING UNIT I

STANSAAB

E 07150 2000 3 | 4 sh | 1

MPU I
REV: 1 " 14

This document must not be copied, reprinted or disclosed to any third party, either wholly or in part without our consent. Any infringement will be prosecuted.
STANSAAB elektronik ab LINKÖPING SWEDEN



Denna handling får ej utan vårt medgivande bekanntgöras kopieras mångfaldigas eller eljest obehörigen utnyttjas. Överträdelse beivras med stöd av lagen om upphovsrätt.
STANSAAB Elektronik AB

OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!

This document must not be copied, reprinted or disclosed to others without our written permission. Any infringement will lead to legal proceedings.
STANSAAB Elektronik AB SWEDEN

REG-NR REG. NO			
KONTR CHECK	SB	GODK. APPROV	0266
SKALA SCALE		KOPIER COPIED	8839 BAo
UTFÖRD ISSUED	9395 SB		740308

MPU I
MICRO PROCESSING UNIT

STANSAAB

E07150 2000 2 | 4 sh | 2

FÖRST ANVAND I FIRST USED ON

INGÅR ÄVEN I ALSO USED ON

REV: 11

R24-26,30-50=1,2k

Oscillator

+12V — L1 330 µH — C3 100 nF — 0V

R2 100k — 2N2222 Y1 — +5V

12,288 MHz X1 — C1 100pF — 2N2369 Y2

C2 18pF — R1 39k — R3 2,2k — R4 150Ω — 0V

12,288 MHz — +5V — R25 — R26 — +5V

6,144 MHz — T = 81,5 ns

Clock divider

7490 — 4T — 10T — 80T — 1280T — 76 kHz

P1 P2

	8	3,07 MHz	
	9	153,6 KHz	MCA
	10	9600 Hz	MCA
	80	1200 Hz	PA
	79	300Hz	PA
	56	37 Hz	
	74	0,6 Hz	
	13	2,3 Hz	
	58	4,6 Hz	
	57	9,2 Hz	

MPUII MEMA changed — 38 — 67

MPUII Read MEMO — 69

Stop CP counter — 9 — 1/sh4

COP halt — 7

74193 — CP-counter — CP2 — CP1 — CP0

I/O address decoder — 74155

Arg ◄ 9/sh4

OP ◄ 10/sh4

78	I/O address 00group	
77	" 04 "	
76	" 08 "	
75	" 0C "	
17	" MUX "	
16	" EBA "	
15	" MCA "	
14	" GEA "	
67	" a0 "	
68	" a1 "	

30 CP2

CP1 ◄ 11/sh4, 12/sh4

21 CP0

11 CP0

13/sh4

PA interrupt — 87 7

GEA interrupt — 88 6

KB interrupt — 89 5

EBA — EBA interrupt — 86 3

MCA — I trans ready — 85 2

MCA — I rec ready — 84 1

Interrupt register

R30 R32 R31 R21 R22 R23 +5V

Interrupt register

27 EBA interrupt req — EBA

General reset (GR) — 18

MPUII Disable interrupt — 35

MPUII Mie3 — 33

Decoder — 7442 1-OF-10

Priority encoder — 74148

Priority logic

Program running level 0 — 40

Program running level 1 — 39

Interrupt request to level 0

Interrupt request

37 Mie 3 — MPU

31 Mie res pri (18) — MPU

29 Set IC MEMA — MPU

29 Select IC MEMA — MPU

36 Mild5 — MPU

2/sh4 — Mi:7 5 — Mi:6 — Mi:3

3/sh4 — CONROM

MPUII Dzero FF — 70

MPUII ALU A=B — 71

4/sh4 — Mia

MPUII AR MC0 — 72

MPUII AR MC1 — 73

74157

Interrupt address register — 74174 — 74175

Interrupt ◄ 14/sh4

15/sh4

32 Interrupt — MPU

6 Interrupt address 1 — MPU

8 " " 2 — MPU

34 " " 3 — MPU

5/sh4 ◄ CP end

MPUII D carry FF — 66

Zero FF — Carry FF — Inhibit clock interrupt

6/sh4 ◄ Carry

7/sh4 ◄ Zero

MPUII Carry — 72

MPUII Zero — 73

MPUII Mie res carry (28) — 64

MPUII Mie res zero (27) — 65

MPUII AR o par — 68

MPUII AR e par — 69

Timer — Inhibit time out — 74164 8-BIT

Time out

GR intern

Reset circuits — +12V — 2N2907A Y2 — 2N 2222 Y1 — Z1 8,2V — C1 47µ

12 +12 V monitor — EBA

11 Reset button

Clock interrupt — 7

MPUII Mie reset (9) — 6

8/sh4 ◄ ARC parity

74175 4-BIT

INGÅR ÄVEN / ALSO USED ON

FÖRST ANVÄND / FIRST USED ON

KONTR CHECK	9510 SB	GODK APPROV	9522 SB
SKALA SCALE		KOPIER COPIED	BA 0
UTFÖRD ISSUED	9595 SB	730409	

REG-NR REG. No

MPU I

MICRO PROCESSING UNIT I

STANSAAB

E07150 2000 1

4 sheet 3

Denna handling får ej utan vårt medgivande besiktigas, kopieras eller på annat sätt mångfaldigas eller utnyttas. Överträdelse beivras enligt gällande lagstiftning.
STANSAAB äger rätten ta

This document must not be copied, reprinted or disclosed to any third party, either wholly or in part, without our consent. Infringement will be prosecuted.
STANSAAB äger rätten ta SWEDEN

This document may not be copied, reprinted or disclosed to a third party without our consent.
Any infringement will be prosecuted.
STANSAAB elektronik AB SWEDEN

P1 P2

MPU II IR MC I 38
MPU II IR MC O 39
MEM MEM:0 59
" " 1 60
" " 2 61
" " 3 62
" " 4 63
" " 5 64
" " 6 65
" " 7 66

Instruction register IR

MPU II OP:3 62 3
MPU II OP:2 63 2

Selector 1 — 74157
Selector 3 — 74157
Mir gates — 74157
Condition selector — 74150

CONROM
ROM 32x8 BIT IM 5600 — 18
IM5600 — 19

Adder
7483 — 45
7483 — 66

Selector 2
74153 — 21
74153 — 9
74153 — 17
Selector 2
74153 — 5

Micro instruction counter MIC — 74198 — 34
For Mi stepper — 31

Mi ROM
ROM 256x4 BIT IM 5603 — 27
IM 5603 — 28

Type decoder — 4-BIT, Fri 74278 — 39
Latches — 74157 — 15
74157 — 40

End decoder
Mij decoder

Arg
Carry
Zero
ARC parity
CF
CT
Mir
Mis
Mij
OV
+5V
R38 1,2k
R34 1,2k

Stop CP counter → 1/sh 3
CP end → 5/sh3
Interrupt → 14/sh3
Interrupt → 15/sh3
Mij: 3,5-7 → 2/sh3
CONROM 2-4 → 4/sh3
9/sh3
10/sh3 OP
13/sh3 CPO
11/sh3 CPI
12/sh3 CPO
4/sh3

P1 P2

71	AR data:7	MPU II
77	C error	MCA
76	C break	MCA
75	C overrun MCA	MCA
74	Status: 1	EBA
83	Status: 2	EBA
82	C repeat	
81	C ID data	
80	C overrun DIA	
78	Spare	
37	Printer inoperable	
79	Spare	
23	Arg 0	MPU II
24	" 1	—"—
25	" 2	—"—
26	" 3	—"—
27	Mark (carry)	MPU II
28	Mod carry	MPU II
19	CONROM: 0	MPU II
20	" 1	"
21	" 2	"
22	" 3	"
23	" 4	"
24	" 5	"
25	" 6	"
26	" 7	"
30	" B	"
31	" C	"
32	" D	"
33	" E	"
34	" F	"
15	CP mit	MPU II
10	End	MPU II
28	Mij start	COP
17	Mi: 0	MPU II
12	" 1	"
13	" 2	"
14	" 3	"
18	" 5	"
19	" 6	"
20	" 7	"
16	Mi: 0	MPU II

P1 P2

MPU II Mie 22
MPU II Mia 35

INGÅR ÄVEN I ALSO USED ON		REG.-NR REG. No.
FÖRST ANVÄND I FIRST USED ON		
KONTR CHECK	100x APPROX	2,2x0
SKALA SCALE	KOPIER COPIED	8810 8AG
UTFÖRD ISSUED	9305 SB	730409

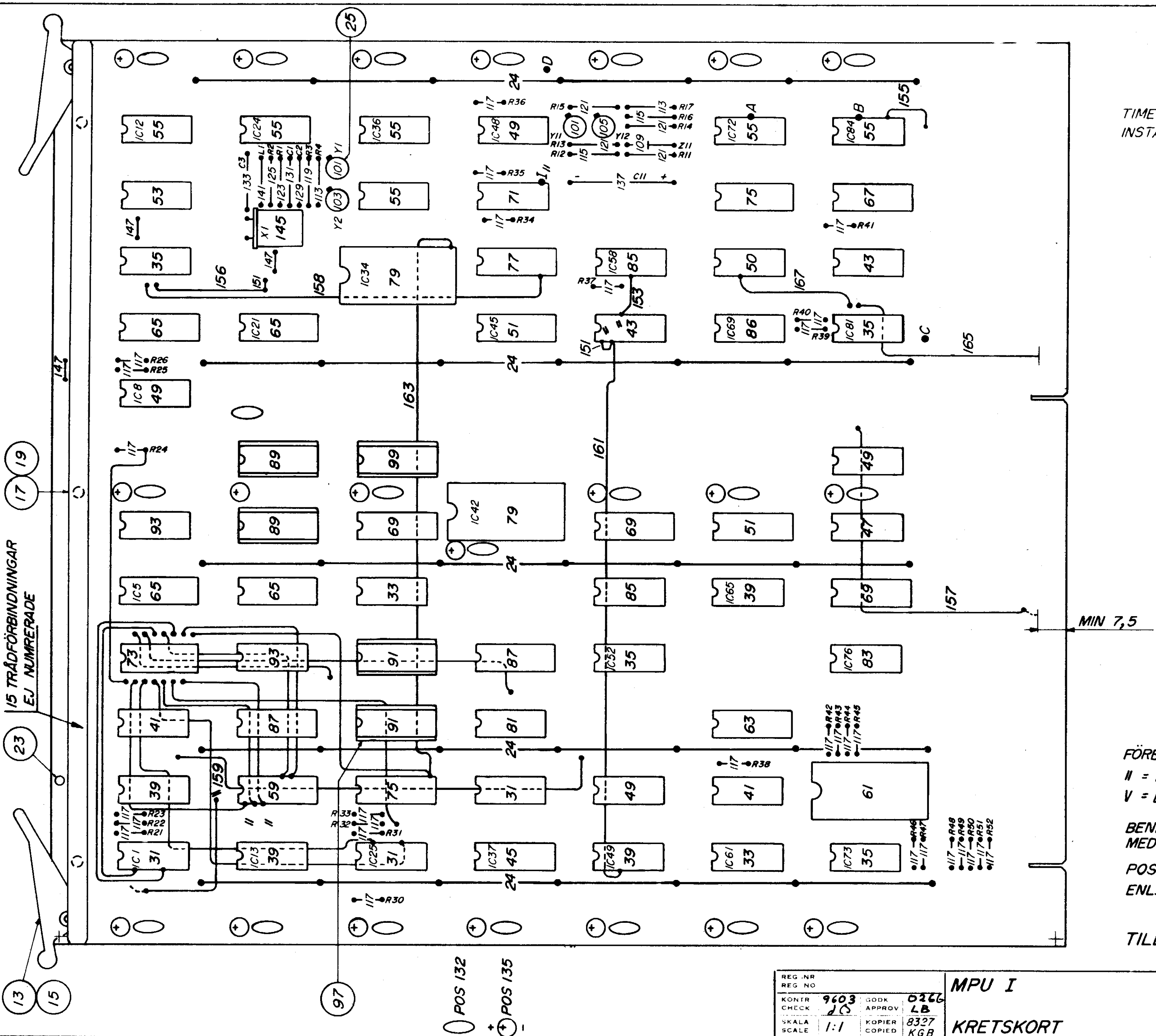
MPU I
MICRO PROCESSING UNIT I

STANSAAB

E07150 2000 1

BLAD SHEET 4 / 4

Rev



TIME OUT KOPPLAS VID INSTALLATION.

FÖRBINDNINGEN BRYTES
II = KOMP.SIDA 4 ST
V = LÖDSIDA SE BLAD 2

BENEN PÅ POS 129 OCH 131 ISOLERAS MED SYSTOFLEX ENL. POS 16

POS 137 OCH 145 FIXERAS MED LIM ENL. F4714

TILLÅTEN LÖDHÖJD 3MM

REG NR / REG NO			
KONTR / CHECK	9603 JO	GODK / APPROV	0266 LB
SKALA / SCALE	1:1	KOPIER / COPIED	8327 KGB
UTFÖRD / ISSUED	9603 JB		73 12 13

MPU I

KRETSKORT
PRINTED BOARD ASSEMBLY

STANSAAB

E07150 0000 3

2 BLAD 1

REV. 1 740206 2 740306 3 740516 4 740905 5 6 7 8 9 10 11 12 E07150 0000 3

MPU I

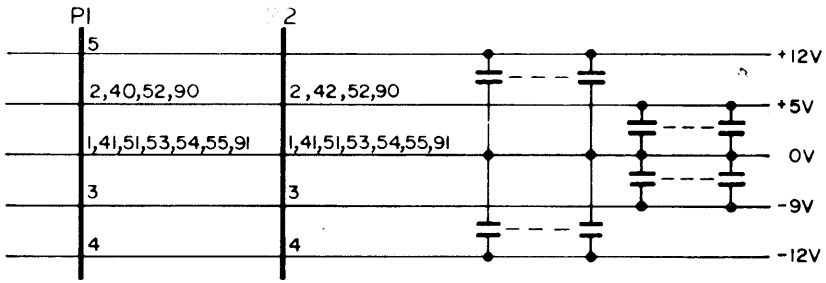
FÖRST ANVÄND I / FIRST USED ON — INGÅR ÄVEN I / ALSO USED ON

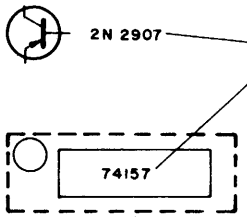
Denna handling får ej utan vårt medgivande bekantgöras kopieras mångfaldigas eller eljest obehörigen utnyttjas. Överträdelse beivras med stöd av gällande lag.
STANSAAB Elektronik AB

OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!

This document must not be copied, reprinted or disclosed to any third party, either wholly or in part, without our consent. Any infringement will lead to legal proceedings.
STANSAAB Elektronik AB SWEDEN

TEST POINTS	
NR	SIGNAL
1	Reg block 0
2	MEM to B Mia
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	
32	
33	
34	
35	
36	
37	
38	
39	
40	
41	





REVISION OF DIAGRAMS AND ASSEMBLY DRAWING -

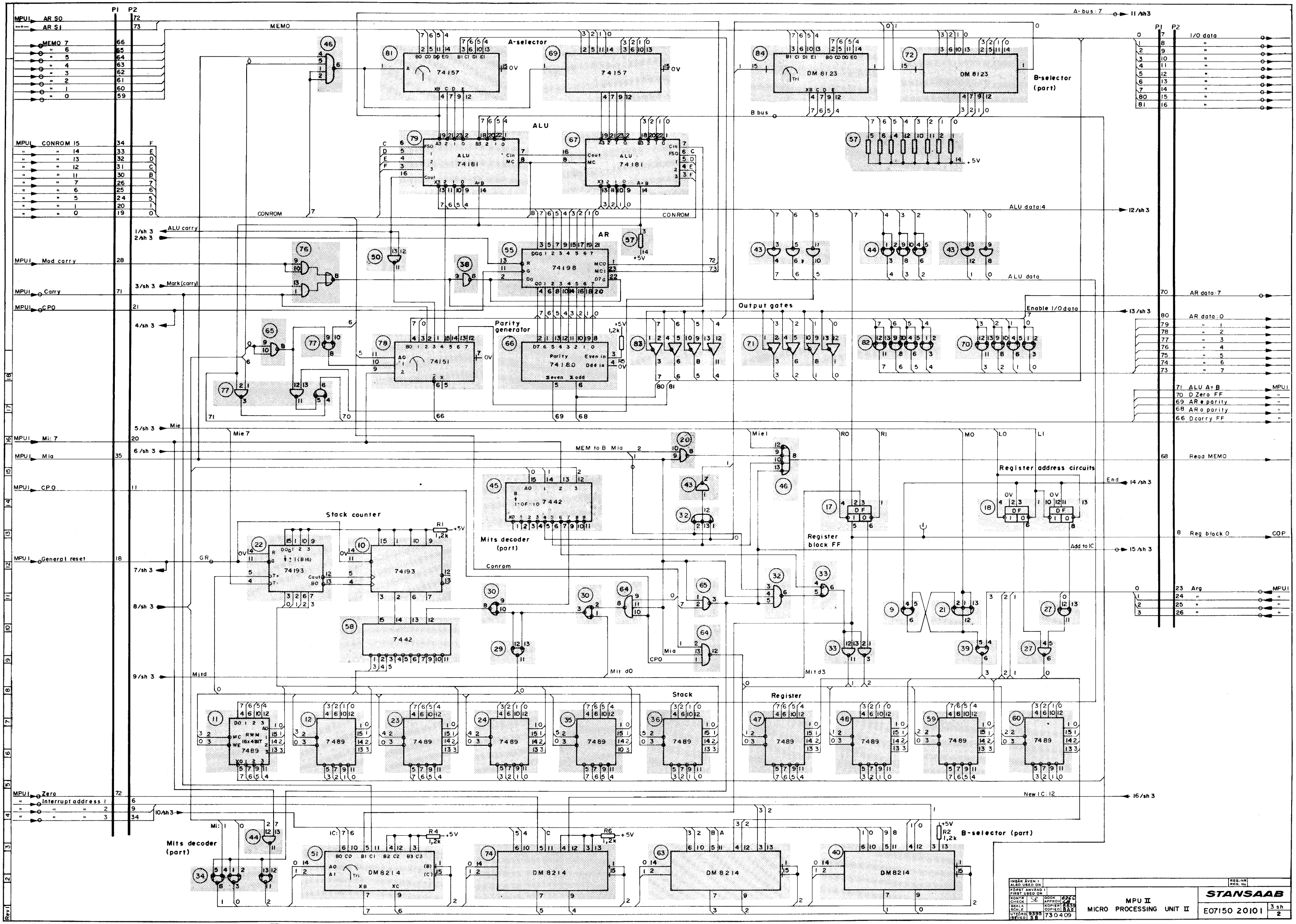
DIAGRAM AND DRAWING NUMBER	SH.	STATUS MOD.	6
E 07150 2010 3	1	REV.	
— " — 1	2	"	
— " — 1	3	"	
		"	
		"	
		"	
		"	
		"	
		"	
E 07150 0010 3		"	5

INGÅR AVEN I ALSO USED ON		REG. NR REG. No	
FÖRST ANVAND I FIRST USED ON			
KONTR CHECK SB	GODK APPROV	MPU II MICRO PROCESSING UNIT II	STANSAAB
SKALA SCALE	KOPIER COPIED 0673 L-D		E 07150 2010 3 — 3sh — 1
UTFÖRD 9395 ISSUED SB	730409		

MPU II — REV. 1 — 1

Denna handling får ej utan vårt medgivande bekantgöras, kopieras eller utlämnas till obehörig eller på annat sätt användas till skada för STANSAAB elektronik ab.

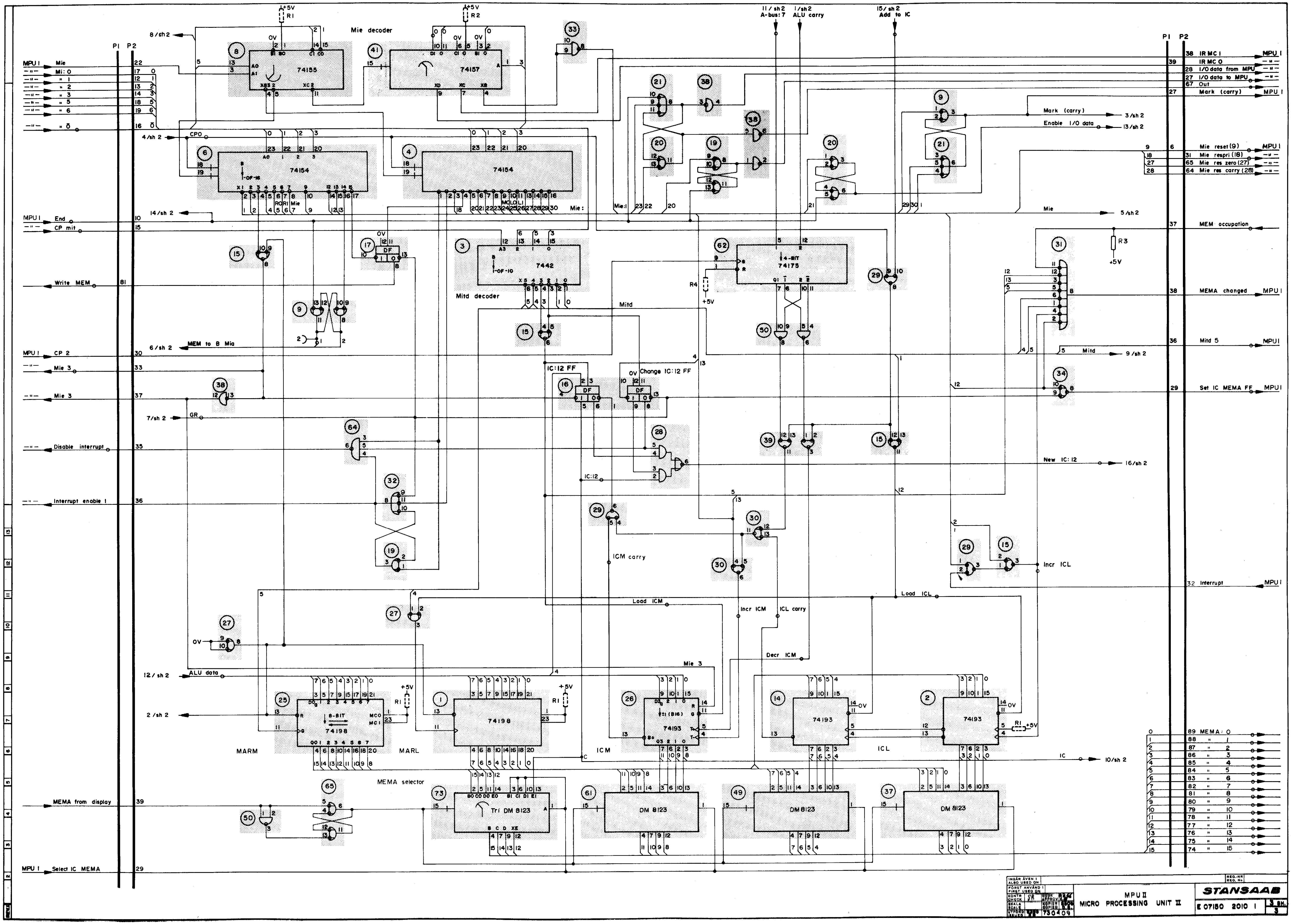
This document must not be copied, reprinted or disclosed to any third party, either wholly or in part, without our consent. Any infringement will lead to legal proceedings.
STANSAAB elektronik ab SWEDEN



Denna handling får ej utan vart medgivande kopieras, utlämnas till tredje man eller användas för tillverkningsändamål. Intrång beivras med stöd av lagstadgade bestämmelser.

This document must not be copied, reprinted or disclosed to a third party nor used for manufacturing purposes. Any infringement will lead to legal prosecution.

INGÅR ÄVEN I ALSO USED ON				REG.NR REG. No.	
FÖRST ANVÄND I FIRST USED ON				STANSAAB	
KONTR CHECK	SB	GODK APPROV	8226	MPU II MICRO PROCESSING UNIT II	E07150 20101 3 sh 2
SKALA SCALE		KOPIERT COPIED	8239		
UTFÖRD ISSUED	9395 SB		730409		



STANSAAB
E 07150 2010 1
MPU II
MICRO PROCESSING UNIT II

Denna handling får ej utan vårt medgivande bekantgöras kopieras mångfaldigas eller ej jest obehörigen utnyttjas. Överträdelse bevisas med stöd av gällande lag.
STANSAAB Elektronik AB

OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!

This document must not be copied, reprinted or disclosed to any third party, either wholly or in part, without our consent. Any infringement will lead to legal proceeding.
STANSAAB Elektronik AB SWEDEN

II MPU

REV.	740206	740327	740517	740813	740911
	2	3	4	5	6

FÖRST ANVAND I / FIRST USED ON

INGÅR ÄVEN I / ALSO USED ON

POS 81 POS 83

FÖRBINDNINGEN BRYTES
II = KOMP. SIDA 5 ST
V = LÖDSIDA SE BLAD 2

TILLÅTEN LÖDHÖJD 3MM

REG.NR / REG. NO			
KONTR / CHECK	JB	GODK / APPROV	8266 LB
SKALA / SCALE	1:1	KOPIER / COPIED	8327 KGB
UTFÖRD / ISSUED	9603 JB		73 12 10

MPU II

KRETSKORT
PRINTED BOARD ASSEMBLY

STANSAAB

E07150 0010 3

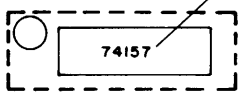
2 BLAD 1

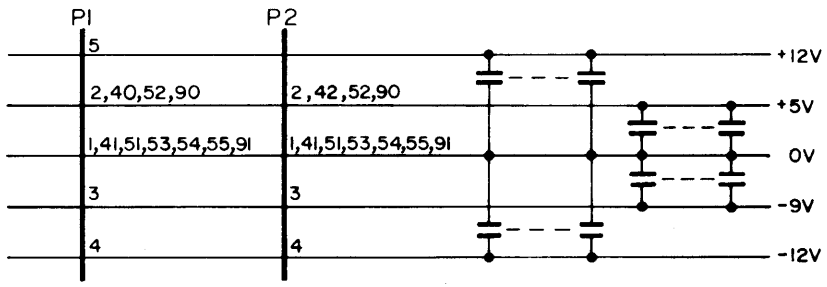
Denna handling får ej utan vårt medgivande bekantgöras, kopieras, mångfaldigas eller eljest obehörigen utnyttjas. Överträdelse beivras med stöd av gällande lag.
STANSAAB elektronik ab

TEST POINTS	
NR	SIGNAL
1	CE 2 RWM
2	CE 1 RWM
3	
4	
5	CS 15 ROM
6	CS 23 ROM
7	CS 7 ROM
8	
9	
10	CS 14 ROM
11	CS 22 ROM
12	CS 12 ROM
13	CE RWM 2000
14	CE RWM 1800
15	
16	CS 5 ROM
17	CS 13 ROM
18	CS 21 ROM
19	ADDRESS DATA SELECTOR BIT 1
20	CS 16 ROM
21	CS 4 ROM
22	CS 20 ROM
23	CS 12 ROM
24	ADDRESS DATA SELECTOR BIT 3
25	ADDRESS DATA SELECTOR BIT 2
26	CS 11 ROM
27	CS 19 ROM
28	CS 3 ROM
29	CS 0 ROM
30	CS ROM 0-7
31	CS ROM 8-15
32	CS 2 ROM
33	CS 18 ROM
34	CS 10 ROM
35	ENABLE ROM/RAM
36	CS 1 ROM
37	CS 17 ROM
38	CS 9 ROM
39	CONSTANT BYTE
40	CS ROM 16-23
41	CS 8 ROM



INDICATED NUMBERS ONLY
EXAMPLES OF EXISTING
CIRCUIT TYPES





REVISION OF DIAGRAMS AND ASSEMBLY DRAWING

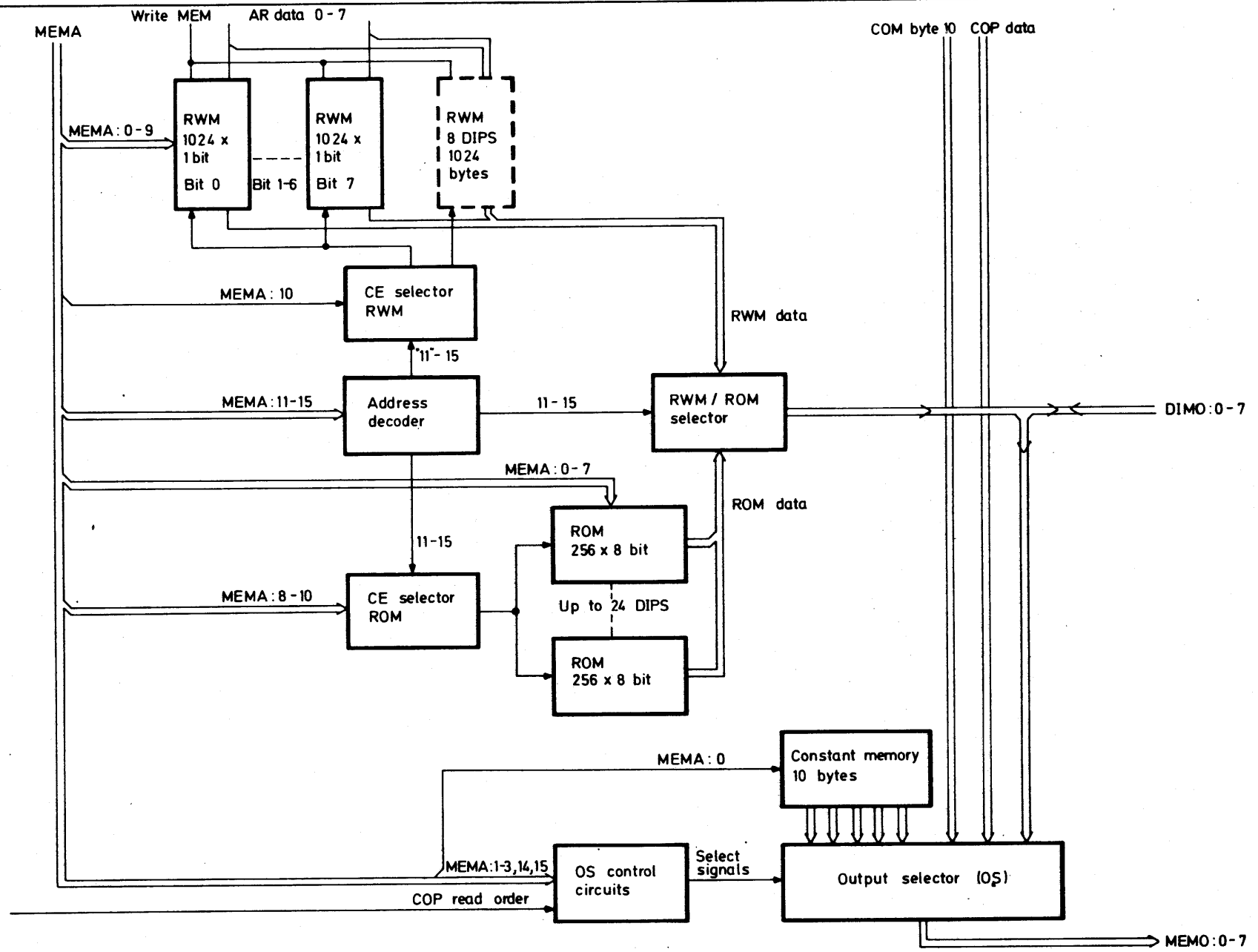
DIAGRAM AND DRAWING NUMBER	SH.	STATUS MOD.	
E 07150 2130 3	1	REV.	1
— " — 3	2	"	
— " — 1	3	"	
		"	
		"	
		"	
		"	
		"	
		"	
E 07150 0130 3		"	

INGÅR ÄVEN I / ALSO USED ON		REG-NR / REG. No	
FÖRST ANVÄND / FIRST USED ON		STANSAAB	
KONTR / CHECK Kal	GODK / APPROV 02 LS	MEM 6/2	
SKALA / SCALE	KOPIER / COPIED 0673 LD		
UTFÖRD / ISSUED 9496 KET	740611	E 07150 2130 3	3 sh 1

This document must not be copied, reprinted or disclosed to any third party either wholly or in part without our consent. Any infringement will lead to legal proceedings.
STANSAAB elektronik ab SWEDEN

MEM 6/2
REV. 741115

Denna handling får ej utan vårt medgivande bekantgöras, kopieras, mångfaldigas eller eljest obehörigen utnyttjas. Överträdelse beivras med stöd av gällande lag.
STANSAAB elektronik ab



REG.-NR REG. NO				MEM 6/2	STANSAAB
KONTR CHECK	Kal	GODK APPROV	0266 LB		E 07150 21303 3 sh
SKALA SCALE		KOPIER COPIED	8506 G.S.		2
UTFÖRD ISSUED	9496 KET	74 06 11			

This document must not be copied, reprinted or disclosed to any third party, either wholly or in part, without our consent. Any infringement will lead to legal proceedings.
STANSAAB elektronik ab SWEDEN

MEM 6/2 REV.1

P1 P2

MPU II
- Write MEM — 31
- AR data: 0 — 30
- " " 1 — 29
- " " 2 — 28
- " " 3 — 27
- " " 4 — 26
- " " 5 — 25
- " " 6 — 24
- " " 7 — 23

- MEMA: 0 — 39
- " " 1 — 38
- " " 2 — 37
- " " 3 — 36
- " " 4 — 35
- " " 5 — 34
- " " 6 — 33
- " " 7 — 32
- " " 8 — 31
- " " 9 — 30
- " " 10 — 29
- " " 11 — 28
- " " 12 — 27
- " " 13 — 26
- " " 14 — 25
- " " 15 — 24

CE Selector ROM

To dipnr: 8 7 6 5 4 3 2 1

To dip no

There are 24 DIP holders
Address inputs and data outputs are connected in parallel. CE inputs are connected to selector (39), (40) and (41) outputs

ROM: 0 1 2 3 4 5 6 7

RWM /ROM data selector

1800

2000

P2

COP read order — 65

+5V

R3

R1

R2

321	
000	Byte 0/1
001	2/3
010	4/5
011	6/7
100	8/9
101	10
110	Data from RWM or ROM
111	Data from COP

P2

COM byte 10: 0 — 14
- " " 1 — 13
- " " 2 — 12
- " " 3 — 11
- " " 4 — 10
- " " 5 — 9
- " " 6 — 8
- " " 7 — 7

COPdata 0 — 73
- " " 1 — 72
- " " 2 — 71
- " " 3 — 70
- " " 4 — 69
- " " 5 — 68
- " " 6 — 67
- " " 7 — 66

Constant memory

Byte:

Bit 0 Bit 1 Bit 2 Bit 3

Bit 4 Bit 5 Bit 6 Bit 7

Output data selector

P1 P2

- MEMO: 0 — 23 — MPU
- " " 1 — 22
- " " 2 — 21
- " " 3 — 20
- " " 4 — 19
- " " 5 — 18
- " " 6 — 17
- " " 7 — 16

- DIMO: 0 — 89 — DIL
- " " 1 — 88
- " " 2 — 87
- " " 3 — 86
- " " 4 — 85
- " " 5 — 84
- " " 6 — 83
- " " 7 — 82

This document, design or the like is the property of Stansaab and may not be copied or communicated without our written permission. All rights reserved in the event of the grant of a patent or the registration of a utility model or design. STANSAAB AKTIEBOLAG PRINTED IN SWEDEN

INGÅR AVEN T. ALSO USED ON		REG.-NR. REG. No.	
FÖRST ANVÄND I FIRST USED ON			
KONSTR. CONSTR.	1200K	APPROV.	
GRANSK. CHECK.	Hal	KOPIER COPIED	
SKALA SCALE		UTGÅR SUPERS.	
UTFÖRD MADE	9486	7312 0 3	

MEM 6/2

STANSAAB

E 07150 2130 1 — 3.sheet 3

This document must not be copied, reprinted or disclosed to any third party, either wholly or in part, without our consent. Any infringement will lead to legal proceedings.
STANSAAB Elektronik AB SWEDEN

OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!

Denna handling får ej utan vårt medgivande bekantgöras kopieras mångfaldigas eller eljest obehörigen utnyttjas. Överträdelse beivras med stöd av gällande lag.
STANSAAB Elektronik AB

MEM 6/2

REV. 1 7409/9 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | E07150 0130 3

FÖRST ANVÄND I / FIRST USED ON

INGÅR ÄVEN I / ALSO USED ON

13 15 23 17 19 55 53

IC1 IC8 IC10 IC11 IC12 IC13 IC20 IC22 IC23 IC30 IC32 IC33 IC34 IC35 IC44 IC46 IC47 IC56 IC58 IC59 IC62 IC68 IC70

POS 59

POS 61

TRÅDBYGEL

TILLÅTEN LÖDHÖJD 3MM

REG NR / REG NO			
KONSTR / CONSTR			
CONTR / CHECK			
SKALA / SCALE	1:1	GODK / APPROV	
KOPIER / COPIED	8327 KGB	UTFÖRD / ISSUED	74 04 10
			9496 KET

MEM 6/2

KRETSKORT
PRINTED BOARD ASSEMBLY

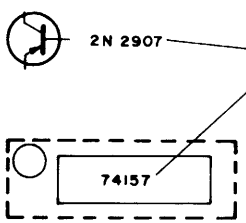
E07150 0130 3

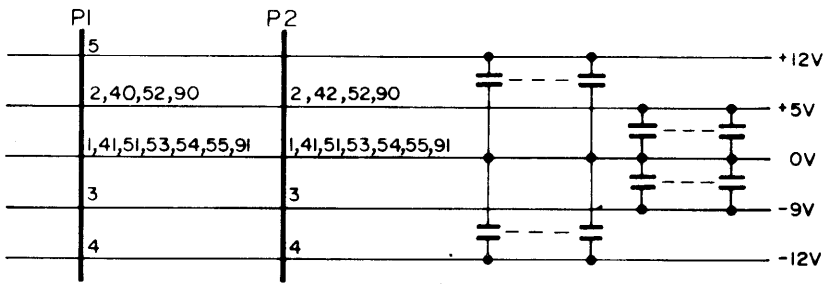
STANSAAB

Denna handling får ej utan vårt medgivande bekantgöras, kopieras, mångfaldigas, överlämnas eller på annat sätt obehörigen utnyttjas. Överträdelse beivras med stöd av gällande lag.
STANSAAB elektronik ab

TEST POINTS

NR	SIGNAL
1	
2	
3	
4	
5	CS 15 ROM
6	
7	CS 7 ROM
8	
9	
10	CS 14 ROM
11	
12	CS 12 ROM
13	START RWM
14	CS ROM 8-15
15	
16	CS 5 ROM
17	CS 13 ROM
18	
19	ADDRESS DATA SELECTOR BIT 1
20	
21	CS 4 ROM
22	
23	CS 12 ROM
24	ADDRESS DATA SELECTOR BIT 3
25	ADDRESS DATA SELECTOR BIT 2
26	CS 11 ROM
27	
28	CS 3 ROM
29	CS 0 ROM
30	
31	
32	CS 2 ROM
33	CS 18 ROM
34	CS 10 ROM
35	ENABLE ROM
36	CS 1 ROM
37	
38	CS 9 ROM
39	CONSTANT BYTE
40	CS ROM 0-7
41	CS 8 ROM





REVISION OF DIAGRAMS

DIAGRAM AND DRAWING NUMBER	SH.	STATUS MOD.
E 07150 2131 3	1	REV.
— " — 4	2	"
— " — 1	3	"
		"
		"
		"
		"
		"
		"
		"

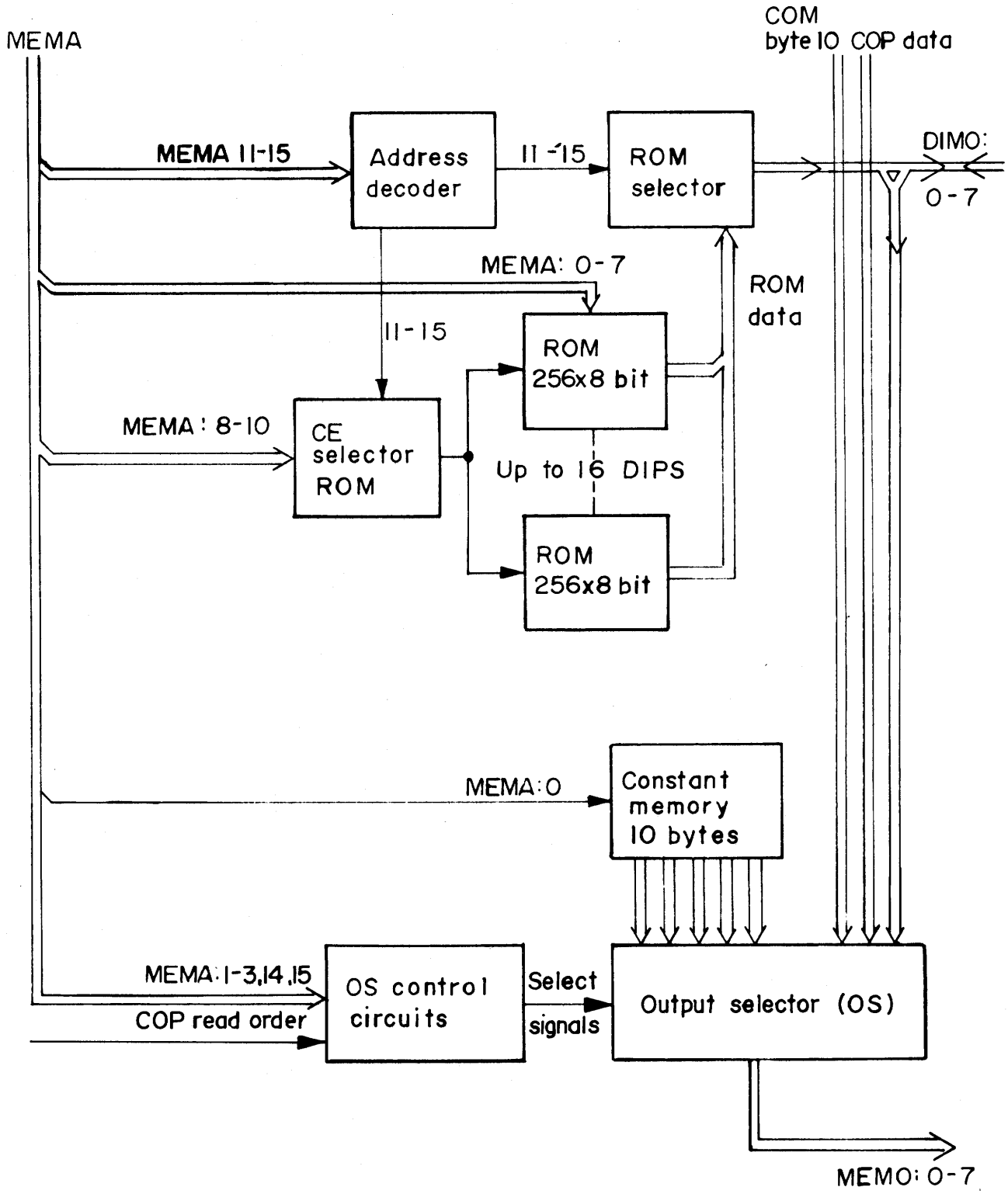
INGÅR AVEN ALSO USED ON FÖRST AN I AND FIRST USED ON	REG NO REG NO		
KONTR 9496 GODK 9496 CHECK KAK APPRO. L-B SKALA KOPIERD 0673 SCALE COPIED L-D UTFÖRD 9496 KET 740611 ISSUED KET	MEM 4/0	STANSAAB E 07150 2131 3	3 sh 1

MEM 4/0
REV.

Denna handling får ej utan vårt medgivande bekantgöras kopieras mångaldigas eller eljest obehörigen utnyttjas. Överträdelse beivras med stöd av gällande lag.
STANSAAB Elektronik AB

OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!

This document must not be copied, reprinted or disclosed to any third party, either wholly or in part, without our consent. Any infringement will lead to legal proceedings.
STANSAAB Elektronik AB SWEDEN



INGÅR ÄVEN I / ALSO USED ON		FÖRST ANVÄND I / FIRST USED ON	
REG.-NR / REG. NO			
KONTR / CHECK	9496 kal	GODK / APPROV	8266 LB
SKALA / SCALE		KOPIER / COPIED	0673 L-D
UTFÖRD / ISSUED	9496 KET	74 06 11	
REV. 1	MEM 4/0	STANSAAB	
		E 07150 2131 4	3 sh / 2

OMJAM BARRAB TERRAB MALMÖ BL. 307:43

P2

MEMA : 0 — 39
- " - 1 — 38
- " - 2 — 37
- " - 3 — 36
- " - 4 — 35
- " - 5 — 34
- " - 6 — 33
- " - 7 — 32
- " - 8 — 31
- " - 9 — 30
- " - 10 — 29
- " - 11 — 28
- " - 12 — 27
- " - 13 — 26
- " - 14 — 25
- " - 15 — 24

CE Selector ROM

There are 16 DIP holders
Address inputs and data outputs are connected in parallel. CE inputs are connected to selector (39) and (40) outputs

(39) 7442 1-OF-10 — To dipnr: 8 7 6 5 4 3 2 1 — 7 12 16 21 28 32 36 29

(40) 7442 — To dip no — 5 10 23 34 41

(37)

(38) 74155 — CI CO BI BO — XC32 1 0 XB0 — A0 A1 — 40 31 30

(47) (36) (37) (36)

IC 1-8, 13, 23-30 — ROM 256x8 BIT 1702 — CE XO — ROM: 0 1 2 3 4 5 6 7

(43) 8123 — BO CO DO EO BI CI DI EI — A B C D XE — 0 1 2 3

(44) 8123 — 4 7 9 12 — 4 5 6 7 — ROM data selector

+5V R12 — (49) (35)

(39)

COP read order — P2 65 — 64 0V — R3 +5V — (68)

+5V R2 — 0V — (59) 74S158 — (19) (25) (24) (2)

+5V R1

321
000 Byte 0/1
001 2/3
010 4/5
011 6/7
100 8/9
101 10
110 Data from RWM or ROM
111 Data from COP

COM byte IO: 0 — P2 14 — 1 13 — 2 12 — 3 11 — 4 10 — 5 9 — 6 8 — 7 7

COPdata 0 — 73 — 1 72 — 2 71 — 3 70 — 4 69 — 5 68 — 6 67 — 7 66

(68)

Constant memory — Byte: — (50) Bit 0 — (51) Bit 1 — (52) Bit 2 — (63) Bit 3 — (64)

Constant memory — Byte: — (65) Bit 4 — (54) Bit 5 — (55) Bit 6 — (66) Bit 7 — (67)

Output data selector

R1 - R12 = 1,2 k

P1 P2

23 MEMO: 0 — MPU
22 - " - 1
21 - " - 2
20 - " - 3
19 - " - 4
18 - " - 5
17 - " - 6
16 - " - 7

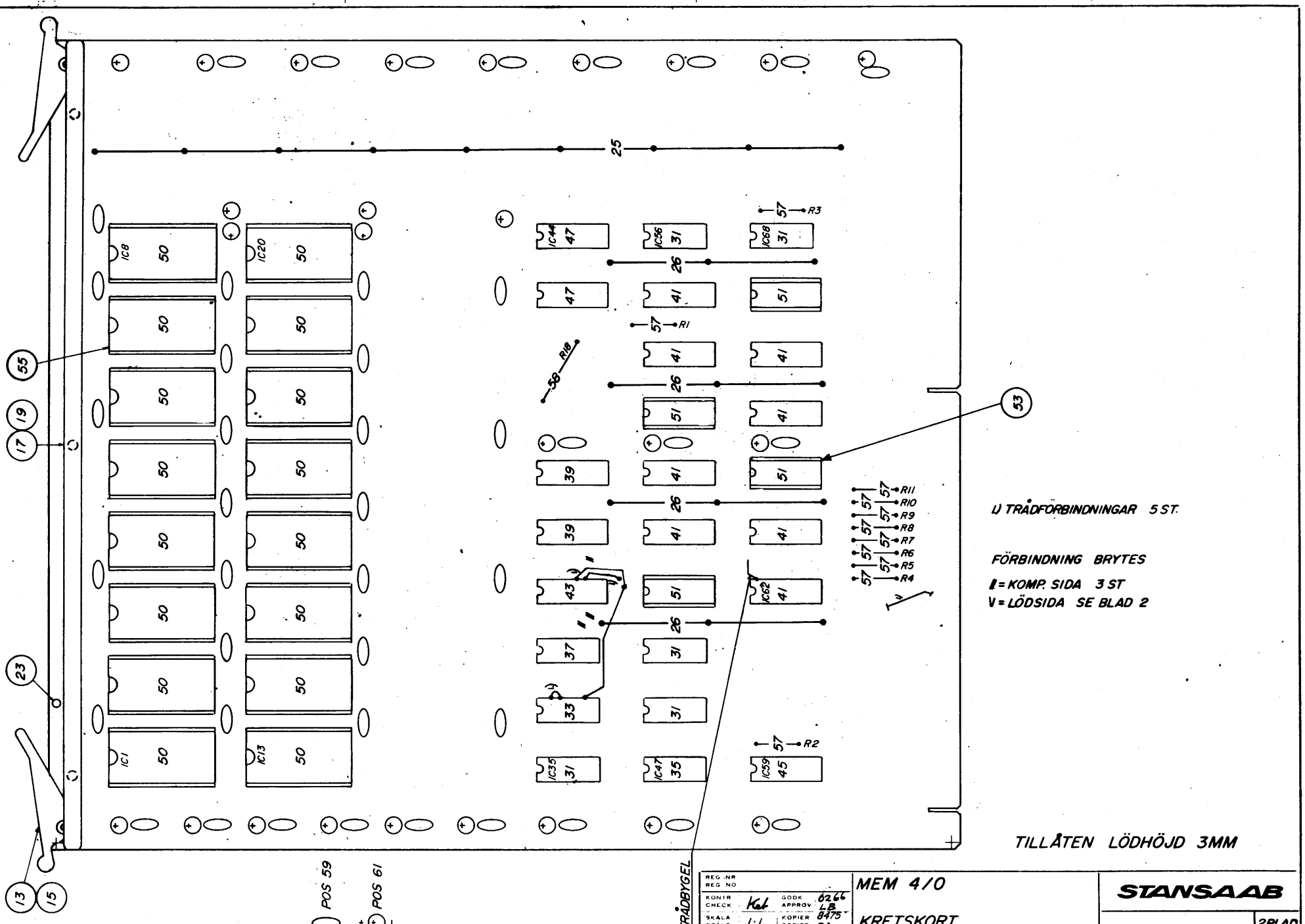
89 DIMO: 0 — MEM/DIL
88 - " - 1
87 - " - 2
86 - " - 3
85 - " - 4
84 - " - 5
83 - " - 6
82 - " - 7

This document must not be copied, reprinted or disclosed to a third party without our written consent. Infringement will lead to legal proceedings.
STANSAAB aktiebolag

Detta dokument får ej utan vårt medgivande kopieras, mångfaldigas eller utlämnas till obehörig. Överträdelse beivras enligt lag.
STANSAAB aktiebolag

REV. I

DRAWN AVEN			
ALSO ANLN			
CHKD ANLN			
FIRST SECT ON			
MONTH	8206		
CHECK			
SCALE			
SCALE			
CUTTING	9406		
ISSUED	REF 731203	MEM 4/0	STANSAAB
		E 07150 2131 1	3 sheet 3



U TRÅDFÖRBINDNINGAR 5 ST.

FÖRBINDNING BRYTES

I = KOMP. SIDA 3 ST

V = LÖDSIDA SE BLAD 2

TILLÅTEN LÖDHÖJD 3MM

This document must not be copied, reproduced or disclosed to any third party, either wholly or in part, without our consent. Any infringement will lead to legal proceedings.
STANSAAB-Elektronik AB SWEDEN

Denna handling får ej utan vårt medgivande lämnas vidare till tredje man eller användas i annat syfte än det för vilket den lämnats ut.
STANSAAB Elektronik AB

OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!

REV.	1	740919											E07150 0131 3

MEM 4/0

FÖRST ANVÄND I / FIRST USED ON

INGÅR ÄVEN I / ALSO USED ON

REG-NR / REG NO			
KONTR / CHECK	Kat	GODK / APPROV	0266 LB
SKALA / SCALE	1:1	KOPIER / COPIED	8475 BS
UTFÖRD / ISSUED	9496 KET		740812

MEM 4/0

KRETSKORT
PRINTED BOARD ASSEMBLY

STANSAAB

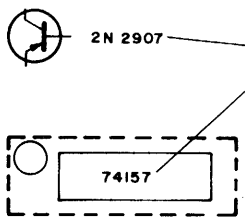
E07150 0131 3

2 BLAD 1

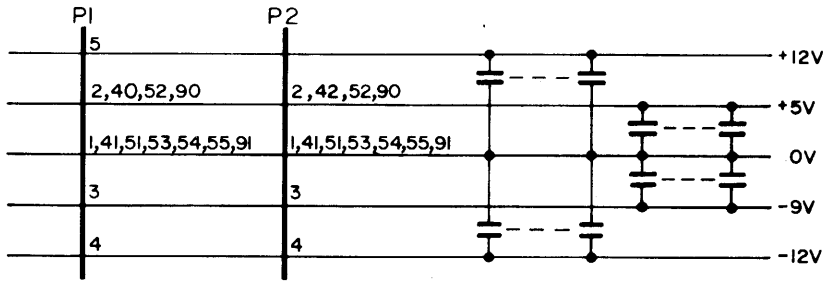
Denna handling får ej utan vart medgivande bekantgöras, kopieras, mångfaldigas eller eljest obehörigen utnyttjas. Överträdelse bevakas med stöd av gällande lag
STAN-AAB elektronik ab

TEST POINTS

NR	SIGNAL
1	CE 0 RWM
2	CE 1 RWM
3	CE 2 RWM
4	CE 3 RWM
5	
6	
7	
8	
9	
10	
11	CS 15 ROM
12	
13	
14	
15	
16	
17	
18	CS 9 ROM
19	CS 10 ROM
20	CS 11 ROM
21	
22	
23	
24	
25	
26	
27	
28	CS 8 ROM
29	CS 12 ROM
30	CS 13 ROM
31	CS 14 ROM
32	CS 5 ROM
33	CS 6 ROM
34	CS 7 ROM
35	CS 4 ROM
36	CS 1 ROM
37	CS 2 ROM
38	CS 3 ROM
39	CS 0 ROM
40	ADDRESS BIT 12
41	ADDRESS BIT 13



INDICATED NUMBERS ONLY
EXAMPLES OF EXISTING
CIRCUIT TYPES



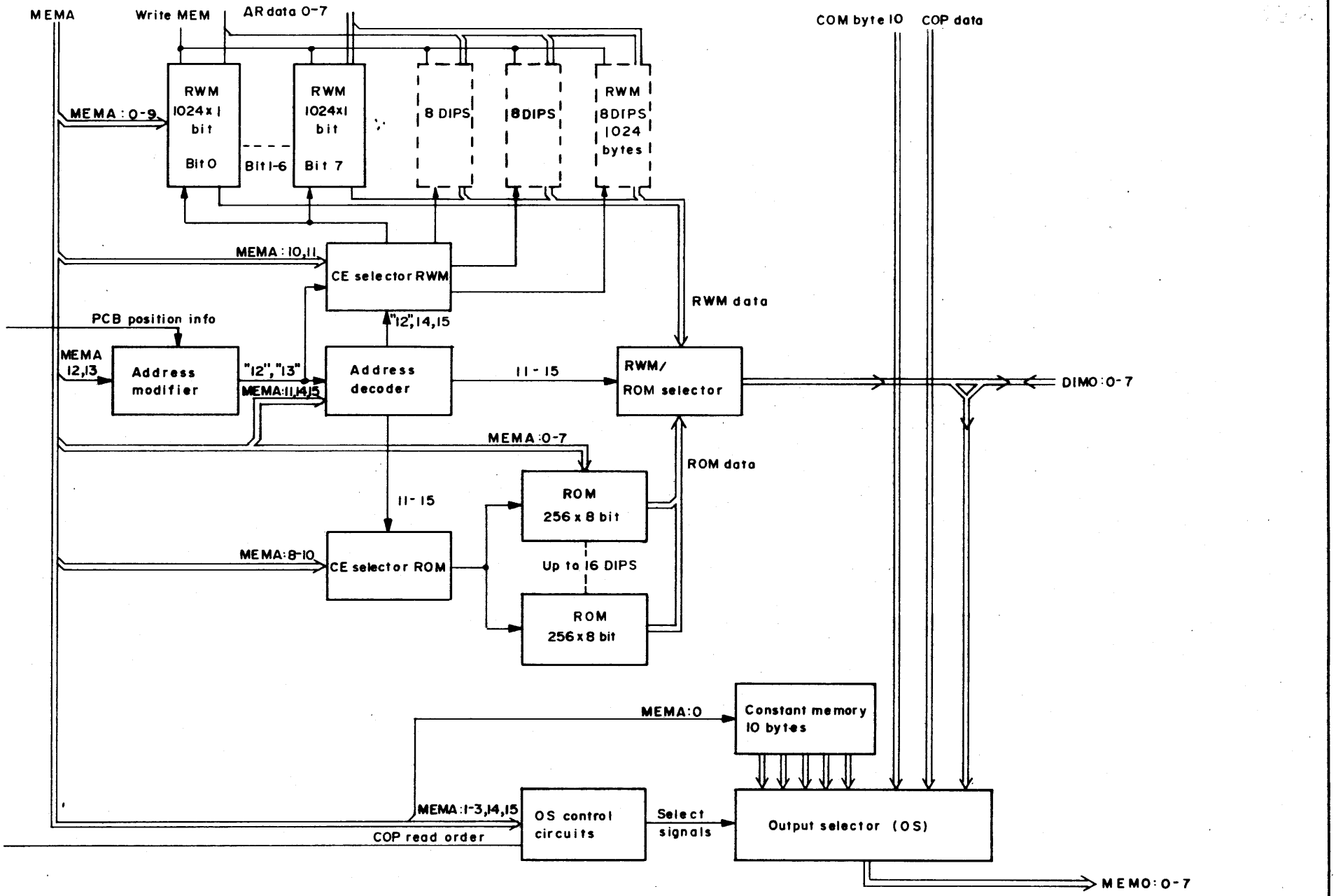
REVISION OF DIAGRAMS AND ASSEMBLY DRAWING

DIAGRAM AND DRAWING NUMBER	SH.	STATUS MOD.	
E 07150 2020 3	1	REV.	1
— " — 3	2	"	
— " — 1	3	"	1
		"	
		"	
		"	
		"	
		"	
		"	
E 07150 0020 3		"	

INGÅR AVEN I ALSO USED ON		REG.-NR REG. No	
FÖRST ANVAND FIRST USED ON		STANSAAB	
KONTR CHECK Kal	GODK APPROV 0266 LB	MEM 4/4	E 07150 2020 3
SKALA SCALE	KOPIER COPIED 0673 L-D		3 sh 1
UTFÖRD ISSUED KET	9406 730320		

This document must not be copied, reprinted or disclosed to any third party, either wholly or in part, without our consent
Any infringement will be dealt with under the law
STAN-AAB elektronik ab SWEDEN

MEM 4/4 REV.1 741115 " 14



REG.-NR REG. NO				MEM 4/4	STANSAAB
KONTR CHECK	Kat	GODK APPROV	0266 LB		
SKALA SCALE		KOPIER COPIED	8839 BAö		E07150 2020 3 — 3 sh
UTFÖRD ISSUED	9496 KET	730320			2

MEM 4/4 Rev I

Address buffer

Data buffer

MPUII Write MEM
AR data: 0 – 7

MEMA: 0 – 15

CE Selector ROM

ROM:

There are 16 DIP holders.
Address inputs and data outputs are connected in parallel.
CE inputs are connected to selector (37) and (38) outputs

To dip no:

Address decoder

Address modifier

PCB position info

COP read order

OS control circuit

CE selector RWM

RWM/ROM data selector

Constant memory

COMbyte 10: 0 – 7

COP data 0 – 7

321	
000	Byte 0/1
001	2/3
010	4/5
011	6/7
100	8/9
101	10
110	Data from RWM or ROM
111	Data from COP

COMbyte 10
COPdata
Byte:
R1 – R8
Bit 0 – Bit 7

Output data selector

MEMO: 0 – 7 MPU

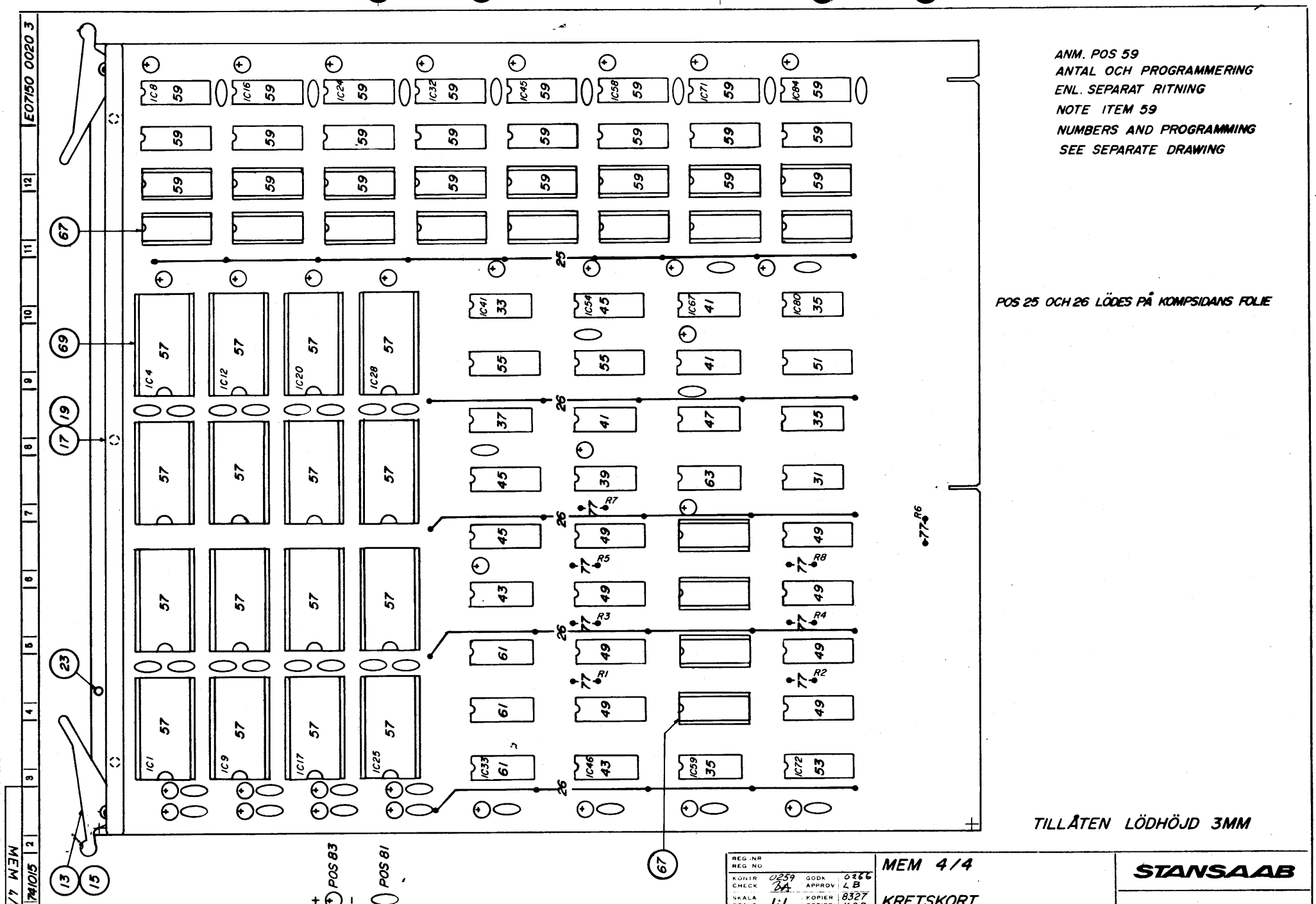
DIMO 0 – 7 MEM/DIL

INGÅR AVEN:	
ALSO USED ON:	
FÖRST ANVÄND:	
FIRST USED ON:	
CONTR	
CHECK	
SKALA	
SCALE	
UTFÄRD	9936
ISSUED	KET 730320

MEM 4/4

STANSAAB

E07150 2020 1 | 3sheet 3



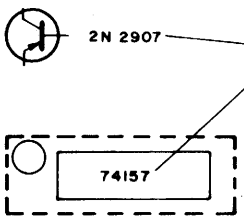
Denna handling får ej utan vårt medgivande bekantgöras kopieras mångfaldigas eller eljest obehörigen utnyttjas. Överträdelse bevakas med stöd av gällande lag.
STANSAAB Elektronik AB

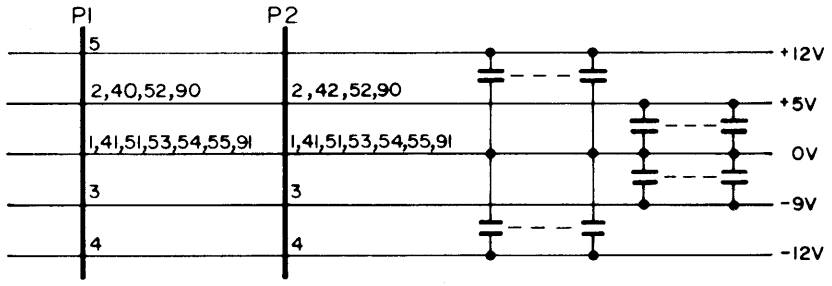
OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!

This document must not be copied, reprinted or disclosed to any third party, either wholly or in part, without our consent. Any infringement will lead to legal proceedings.
STANSAAB Elektronik AB SWEDEN

Denna handling får ej utan vart medgivande bekantgöras, kopieras, mångfaldigas eller eljest obehörigen utnyttjas. Överträdelse beivras med stöd av gällande lag
STANSAAB elektronik ab

TEST POINTS	
NR	SIGNAL
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	CS 15 ROM
12	
13	
14	
15	
16	
17	
18	CS 9 ROM
19	CS 10 ROM
20	CS 11 ROM
21	
22	
23	
24	
25	
26	
27	
28	CS 8 ROM
29	CS 12 ROM
30	CS 13 ROM
31	CS 14 ROM
32	CS 5 ROM
33	CS 6 ROM
34	CS 7 ROM
35	CS 4 ROM
36	CS 1 ROM
37	CS 2 ROM
38	CS 3 ROM
39	CS 0 ROM
40	ADDRESS BIT 12
41	ADDRESS BIT 13





REVISION OF DIAGRAMS

DIAGRAM AND DRAWING NUMBER	SH.	STATUS MOD.
E 07150 2021 3	1	REV.
— " — 4	2	"
— " — 1	3	"
		"
		"
		"
		"
		"
		"
		"
		"

This document must not be copied, reprinted or disclosed to
any third party, either wholly or in part, without our consent.
Any infringement will be legally prosecuted.
STANSAAB elektronik ab SWEDEN

MEM 4/0

REV.

INGÅR AVEN
ALSO USED ON
FIRST ANVAND
FIRST USED ON
KONTR 9496 GODK 0222
CHECK KÖPT. LB
SKALA KOPIER 0673
SCALE COPIED L-D
UTFÖRD 9496 730320
ISSUED KET

REG NR
REG NO

STANSAAB

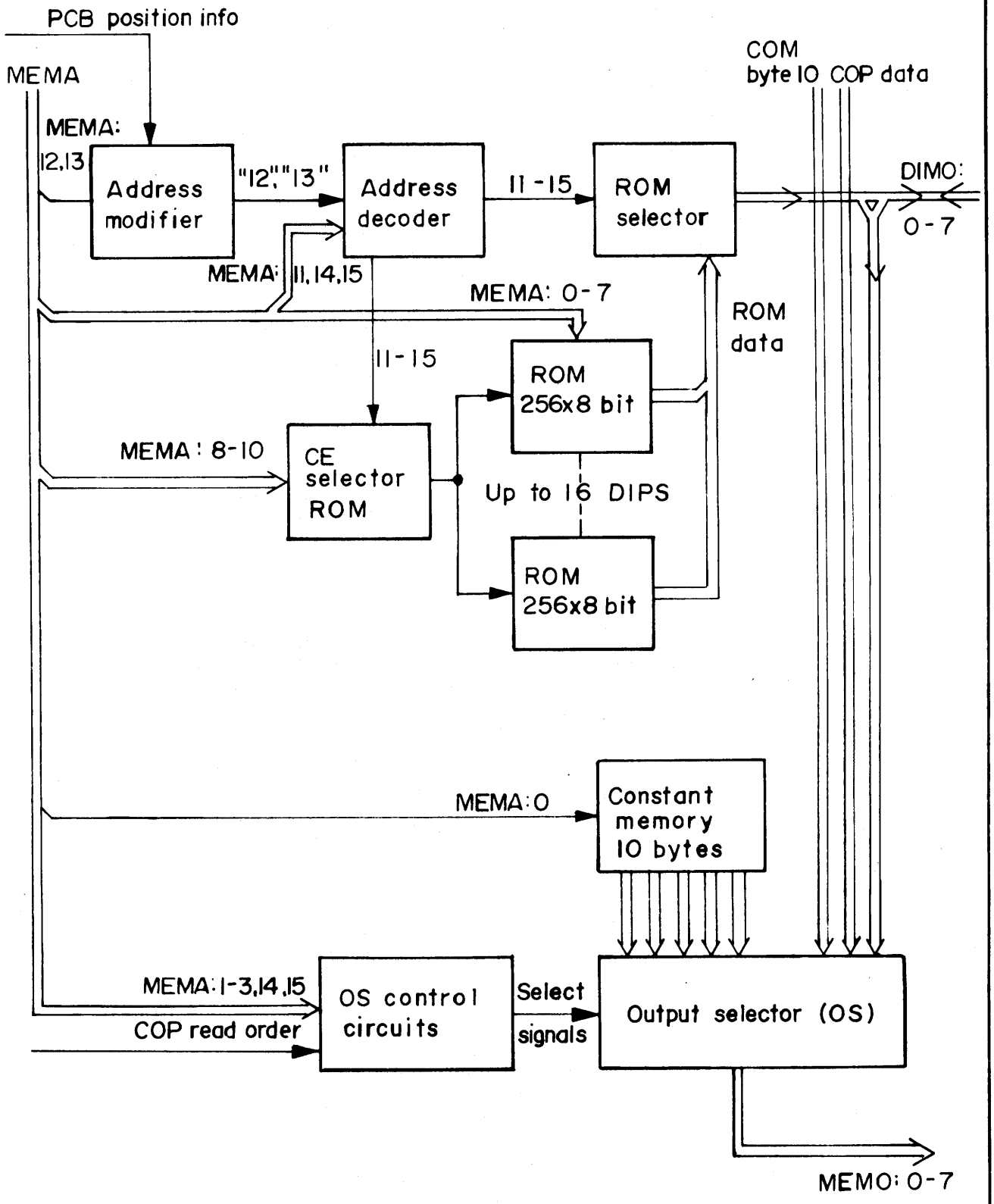
MEM 4/0

E 07150 20213 | 3 sh | 1

Denna handling får ej utan vårt medgivande bekantgöras kopieras mångfaldigas eller eljest obehörigen utnyttjas. Överträdelse beivras med stöd av gällande lag.
STANSAAB Elektronik AB

OM NAGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!

This document must not be copied, reprinted or disclosed to any third party, either wholly or in part, without our consent. Any infringement will lead to legal proceedings.
STANSAAB Elektronik AB SWEDEN



INGÅR ÄVEN I ALSO USED ON				FÖRST ANVÄND I FIRST USED ON	
REG.-NR REG. NO					
KONTR CHECK	9496 Kal.	GODK APPROV	0266 LB	MEM 4/0	STANSAAB
SKALA SCALE		KOPIER COPIED	0673 L-D		E 07150 2021 4 — 3 sh — 2
UTFÖRD ISSUED	9496 KET	730320			
REV. 1					

P2

MEMA: 0 — 39
-"- 1 — 38
-"- 2 — 37
-"- 3 — 36
-"- 4 — 35
-"- 5 — 34
-"- 6 — 33
-"- 7 — 32
-"- 8 — 31
-"- 9 — 30
-"- 10 — 29
-"- 11 — 28
-"- 12 — 27
-"- 13 — 26
-"- 14 — 25
-"- 15 — 24

Address decoder

CE Selector ROM

To dip no: 11 31 30 29 20 19 18 28 — 34 33 32 35 36 37 36 39

IC 1-4, 9-12, 17-20, 25-28: ROM AO 1 2 3 4 5 6 7 / CE 256x8 BIT 1702 / XO 1 2 3 4 5 6 7 / 4 5 6 7 8 9 10 11 / ROM: 0 1 2 3 4 5 6 7

There are 16 DIP holders.
Address inputs and data outputs are connected in parallel.
CE inputs are connected to selector (37) and (38) outputs

RWM/ROM data selector

P2

PCB position info — 62
64 — 0V
PCB position info — 63
COP read order — 65

Address modifier

OS control circuit

COMbyte IO: 0 — P2 — 14
— " — 1 — 13
— " — 2 — 12
— " — 3 — 11
— " — 4 — 10
— " — 5 — 9
— " — 6 — 8
— " — 7 — 7

COP data 0 — 73
— " — 1 — 72
— " — 2 — 71
— " — 3 — 70
— " — 4 — 69
— " — 5 — 68
— " — 6 — 67
— " — 7 — 66

321	
000	Byte 0/1
001	2/3
010	4/5
011	6/7
100	8/9
101	10
110	Data from RWM or ROM
111	Data from COP

COMbyte IO
COPdata
Byte:
Bit 0 Bit 1 Bit 2 Bit 3
Output data selector

Constant memory

COM byte IO
COPdata
Byte
Bit 4 Bit 5 Bit 6 Bit 7
Output data selector.

P1 P2
MEMO: 0 — 23 — MPU
-"- 1 — 22
-"- 2 — 21
-"- 3 — 20
-"- 4 — 19
-"- 5 — 18
-"- 6 — 17
-"- 7 — 16

DIMO 0 — 89 — MEM/DIL
-"- 1 — 88
-"- 2 — 87
-"- 3 — 86
-"- 4 — 85
-"- 5 — 84
-"- 6 — 83
-"- 7 — 82

Denna handling är ej stöv. vård. medgivande bekräftades, lämnas ej utan tillgängas eller göras obehörigen tillgänglig. Överträdelse medför skadeståndsskyldighet. STANSAAB elektronik ab

This document must not be copied, reproduced or disclosed to any third party, either wholly or part, without our consent. Any infringement is subject to prosecution. STANSAAB elektronik ab SWEDEN

REV. I

MEM 4/0

STANSAAB

E 07150 2021 1 — 3sheet 3

Denna handling får ej, utan vårt medgivande bekantgöras kopieras mångfaldigas eller eljest obehörigen utnyttjas. Över trådelse beivras med stöd av gällande lag
STANSAAB Elektronik AB

OM NÅGOT ÄR OKLART: **FRÅGA!**
IF IN DOUBT: **ASK!**

This document must not be copied, reprinted or disclosed to any third party without our consent
Any infringement will lead to legal proceedings.
STANSAAB Elektronik AB SWEDEN

MEM 4/0

REV.			
1	74105	2	741106

FÖRST ANVÄND I
FIRST USED ON

INGÅR ÄVEN I
ALSO USED ON

ANM. POS 59
ANTAL OCH PROGRAMMERING
ENL. SEPARAT RITNING
NOTE ITEM 59
NUMBERS AND PROGRAMMING
SEE SEPARATE DRAWING

POS 25 OCH 26 LÖDES PÅ KOMPSIDANS FOLIE

TILLÅTEN LÖDHÖJD 3MM

POS 83 (+)
POS 81 (−)

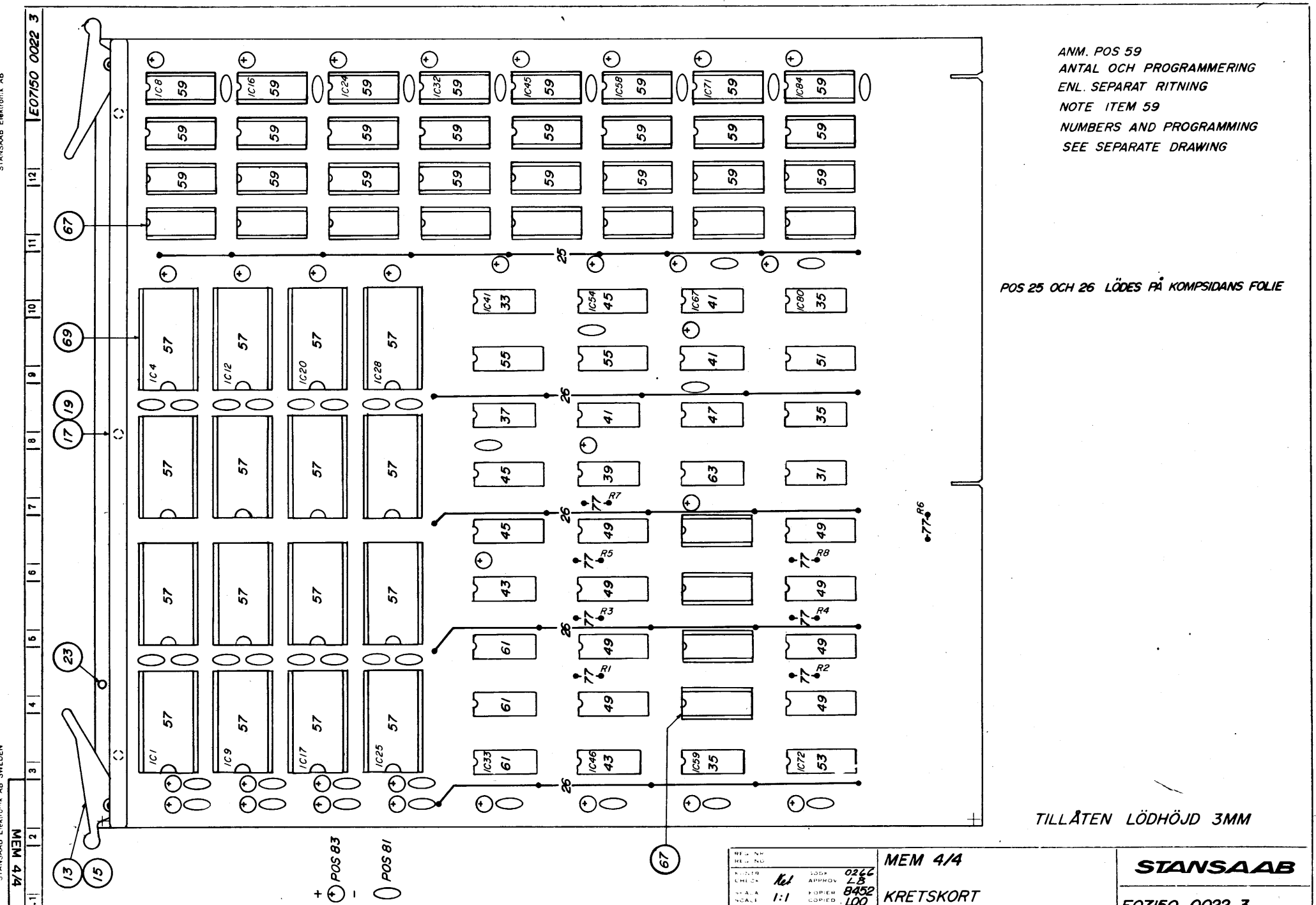
REG. NR / REG. NO			
KONSTR / DRAWN		GODK / APPROV	
CHECK	Kef.		
SKALA / SCALE	1:1	KOPIER / COPIED	8475 BS
DATUM	9496		
SIGN	KET		740812

MEM 4/0

KRETSKORT
PRINTED BOARD ASSEMBLY

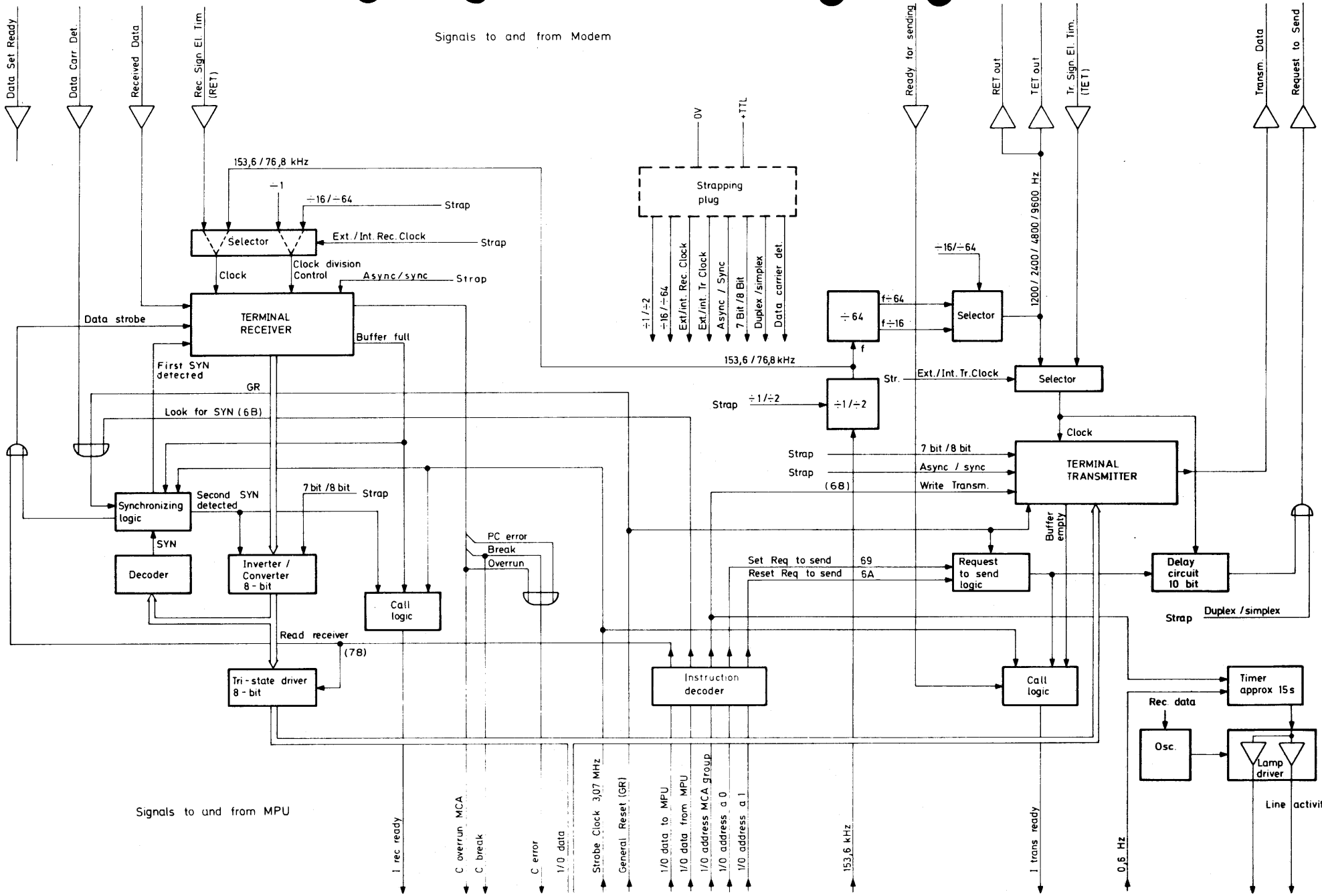
STANSAAB

E07150 0021 3



Denna handling får ej utan vårt medgivande bekantgöras kopieras mångfaldigas eller på annat sätt användas otrofast. Överträdelse beivras med stöd av gällande lag.
STANSAAB Elektronik AB

This document must not be copied, reprinted or disclosed to any third party, either wholly or in part, without our consent. Any infringement will lead to legal proceedings.
STANSAAB Elektronik AB SWEDEN



REG NR / REG NO			
KONTR CHECK	J.B	GODK APPROV	
SKALA SCALE		KOPIER COPIED	8506 GS
UTFÄRD ISSUED	96:3 JB	73 08 16	

MCA
MODEM COMMUNICATION ADAPTER

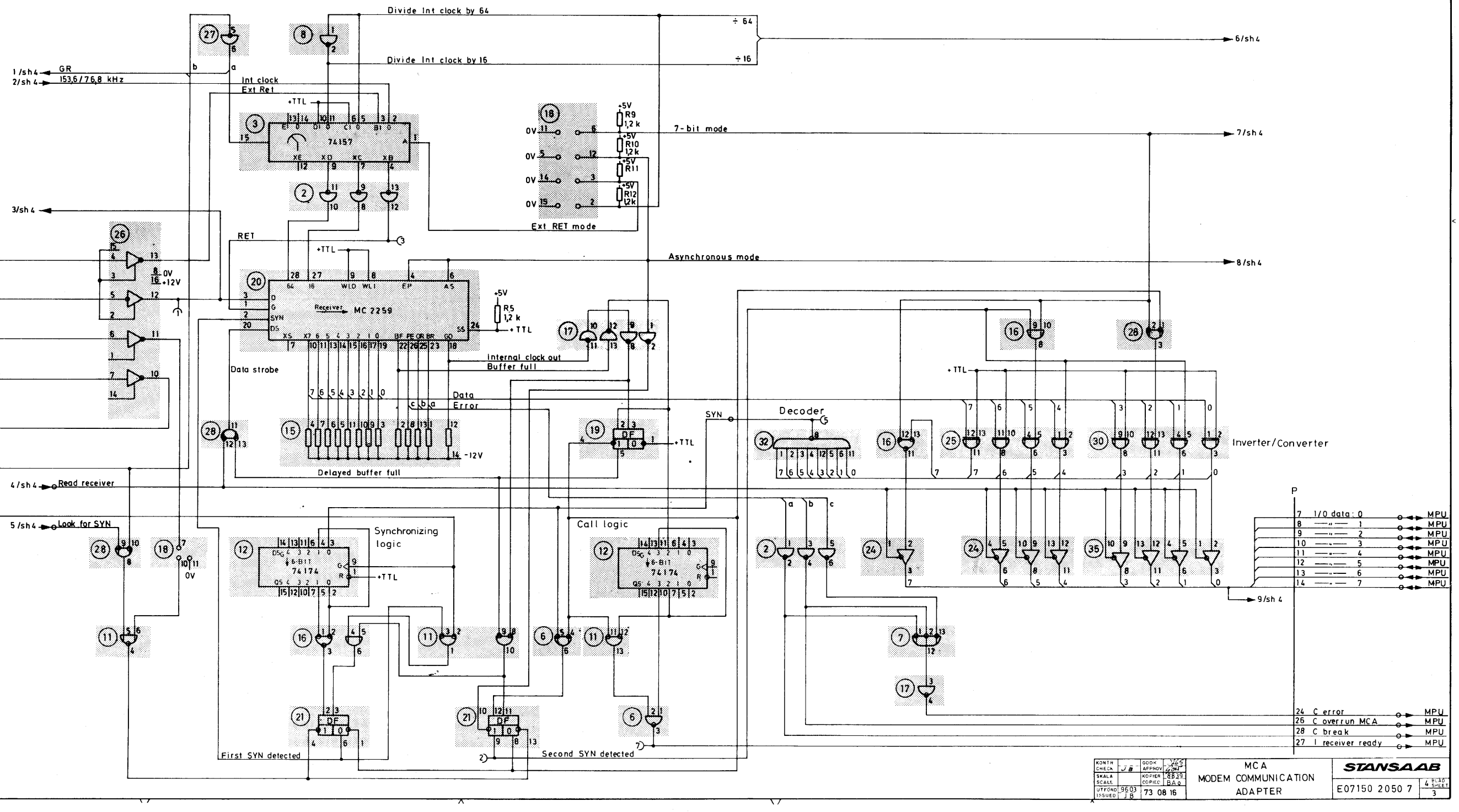
STANSAAB

E 07150 2050 2 | 4 sh | 2

MCA | REV.1 | FÖRST ANVÄND FIRST USED ON | INGÅR ÄVEN I ALSO USED ON

OMJAM BARRIÄT ES 10E JB

Any infringement will lead to legal proceedings
STANSAAB Elektronik AB SWEDEN
IF IN DOUBT: ASK!
STANSAAB Elektronik AB
tråderna behöver med stöd av gällande lag



OM NÅGOT ÄR OKLART: FRÅGA! IF IN DOUBT: ASK!

Denna handling får ej, utan vårt medgivande kopieras, delges eller på annat sätt överlämnas till utomstående. Överträdelse beivras med stöd av gällande lag. STANSAAB Elektronik AB

This document must not be copied, reprinted or disclosed to a third party without our permission and be used for contract. Any infringement will lead to legal proceedings. STANSAAB Elektronik AB SWEDEN

REV.1 (40823)

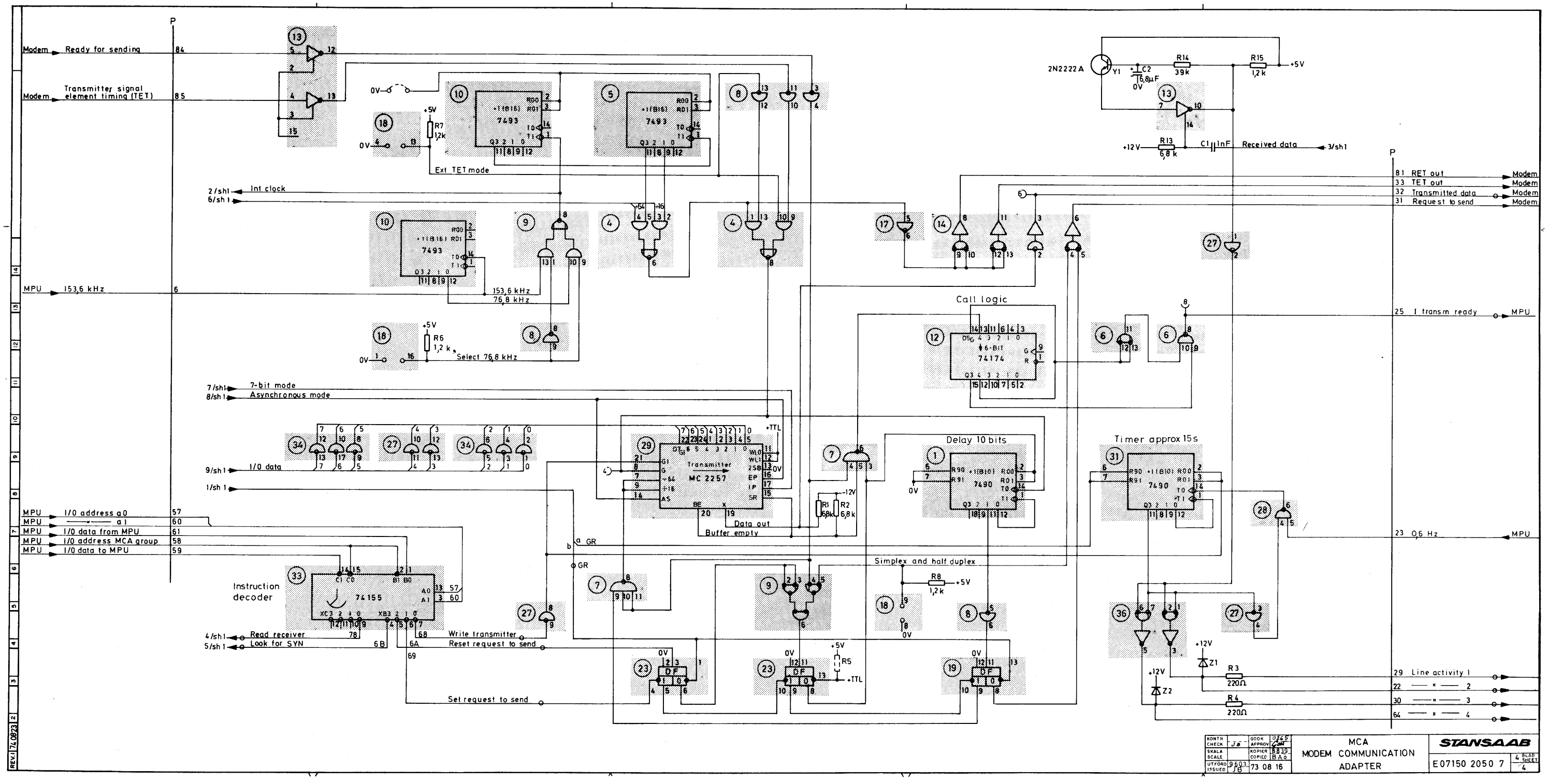
KONTR CHECK	J B	GODK APPROV	
SKALA SCALE		KOPIEN COPIED	88 39 BAO
UTFÄRD ISSUED	9603 J B	73 08 16	

MCA
MODEM COMMUNICATION ADAPTER

STANSAAB

E07150 2050 7

4 SHEET 3



This document must not be copied, reprinted or disclosed to any third party, in part or in whole, without our consent. Any infringement will lead to legal action.
STANSAAB Elektronik AB SWEDEN

Denna handling är ej utan vårt medgivande kopierbar, lämnas ej utomstående eller eljest obehörigen utnyttjas. Överträdelse leder till skadeståndsansvar.
STANSAAB Elektronik AB

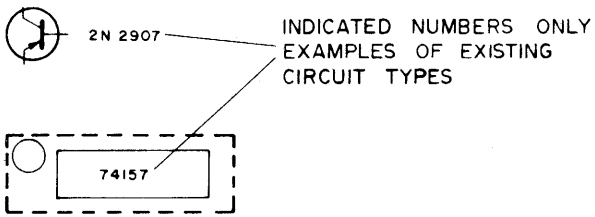
OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!

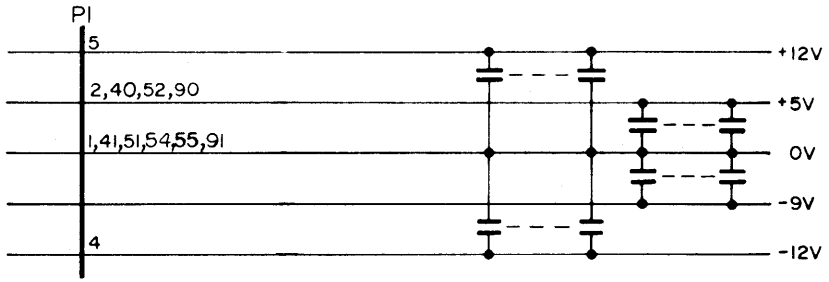
KONTR CHECK	JB	GODK APPROV	0765	MCA MODEM COMMUNICATION ADAPTER	STANSAAB
SKALA SCALE		KOPIER COPIED	8839 BAO		E07150 2050 7
UTFÄRD ISSUED	9603 JB	73 08 16			4 BLAD SHEET 4

REV: 7 (0823)

TEST POINTS

NR	SIGNAL
1	Received Data
2	Second Syn Detected
3	Receiver Clock
4	Transmitter Clock
5	Syn
6	Transmitted Data
7	Receiver ready
8	Transmitter ready
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	
32	
33	
34	
35	
36	
37	
38	
39	
40	
41	





REVISION OF DIAGRAMS AND ASSEMBLY DRAWING

DIAGRAM AND DRAWING NUMBER	SH.	STATUS MOD.	-	1	2	3
E 07150 2060 3	1	REV.	1			
— " — 2	2	"				
— " — 7	3	"		1		
— " — 7	4	"		1		
		"				
		"				
		"				
		"				
		"				
E 07150 0050 3		"		1	2	3

MCA
REV.1 740823
" 14

INGÅR AVEN
A SO USED ON
FIRST AN AND
FIRST USED ON
CONTR
CHECK JB
SCALE
SCALE
DRAWN 9603 JR
ISSUED JR 730816
GODK
APPRO
KOPIER 9977
COPIED L.P.
REG. NR
REG. No

MCA
MODEM COMMUNICTION
ADAPTER

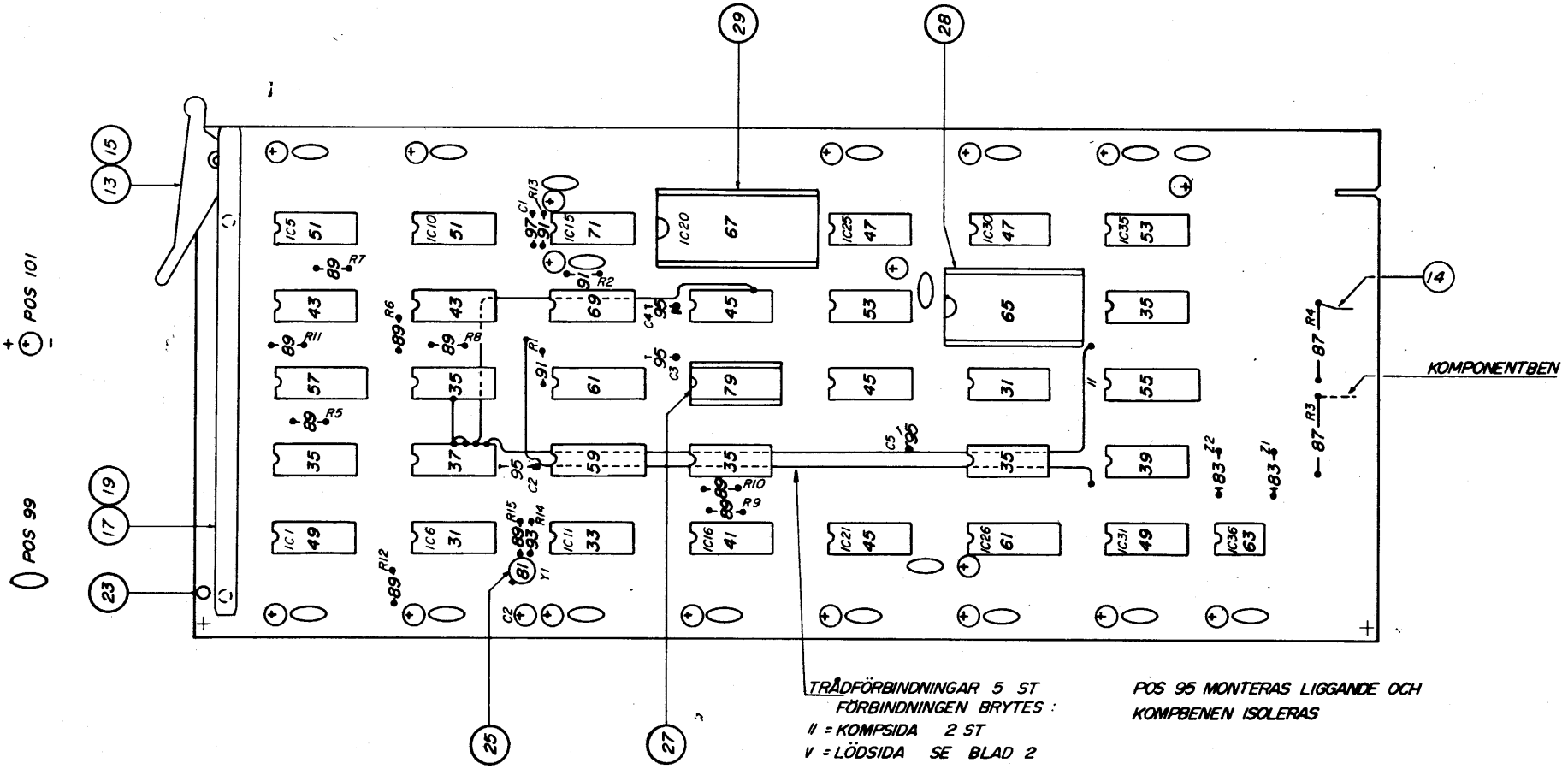
STANSAAB

E 07150 2050 3

4 sh
1

Denna handling får ej utan vårt medgivande bekantgöras kopieras mångfaldigas eller eljest obehörigen utnyttjas. Överträdelse beivras med stöd av gällande lag.
STANSAAB Elektronik AB

OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!



TILLÅTEN LÖDHÖJD 3MM

GÄLLER FÖR KLICHE REV I

REG.NR / REG NO	
KUNTR / CHECK	J.B.
GODK / APPROV	0266 LB
SKALA / SCALE	1:1
KOPIER / COPIED	8327 KGB
UTFÖRD / ISSUED	9603 JB 73 11 16

MCA

KRETSKORT
PRINTED BOARD ASSEMBLY

STANSAAB

E07150 0050 3

2 BLAD 1

REV.	1	2	3	4	5	6	7	8	9	10	11	12
	740103	740509	740517	740823	741107							E07150 0050 3

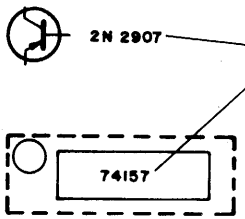
FÖRST ANVAND I / FIRST USED ON

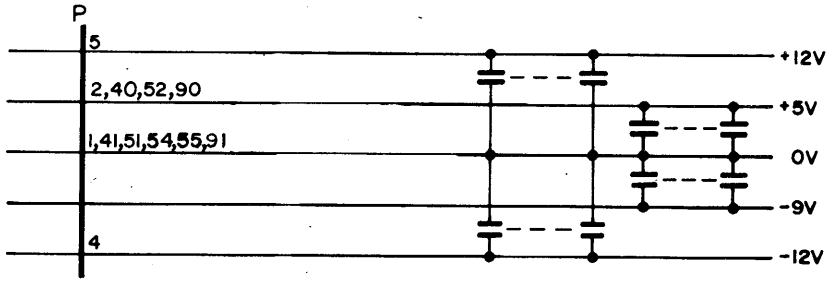
INGÅR ÄVEN I / ALSO USED ON

This document must not be copied, reprinted or disclosed to any third party, either wholly or in part, without our consent. Any infringement will lead to legal proceedings.
STANSAAB Elektronik AB SWEDEN

Denna handling får ej utan, vårt medgivande bekantgöras, kopieras, mångfaldigas eller eljest obehörigen utnyttjas. Överträdelse beivras med stöd av gällande lag.
STANSAAB elektronik ab

TEST POINTS	
NR	SIGNAL
1	DATA ACC. FF
2	DATA VALID
3	TRANSMITT MODE
4	DATA ACCEPTED
5	DATA FROM EBA
6	DATA VALID FF
7	BUS ON
8	DATA TO EDB
9	PARTICIPATION DETERMINED
10	STATUS TO EBA
11	STATUS BIT 2
12	GENERAL CALL RECEIVED
13	WAIT FOR STATUS
14	
15	STATUS BIT 1
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	
32	
33	
34	
35	
36	
37	
38	
39	
40	
41	



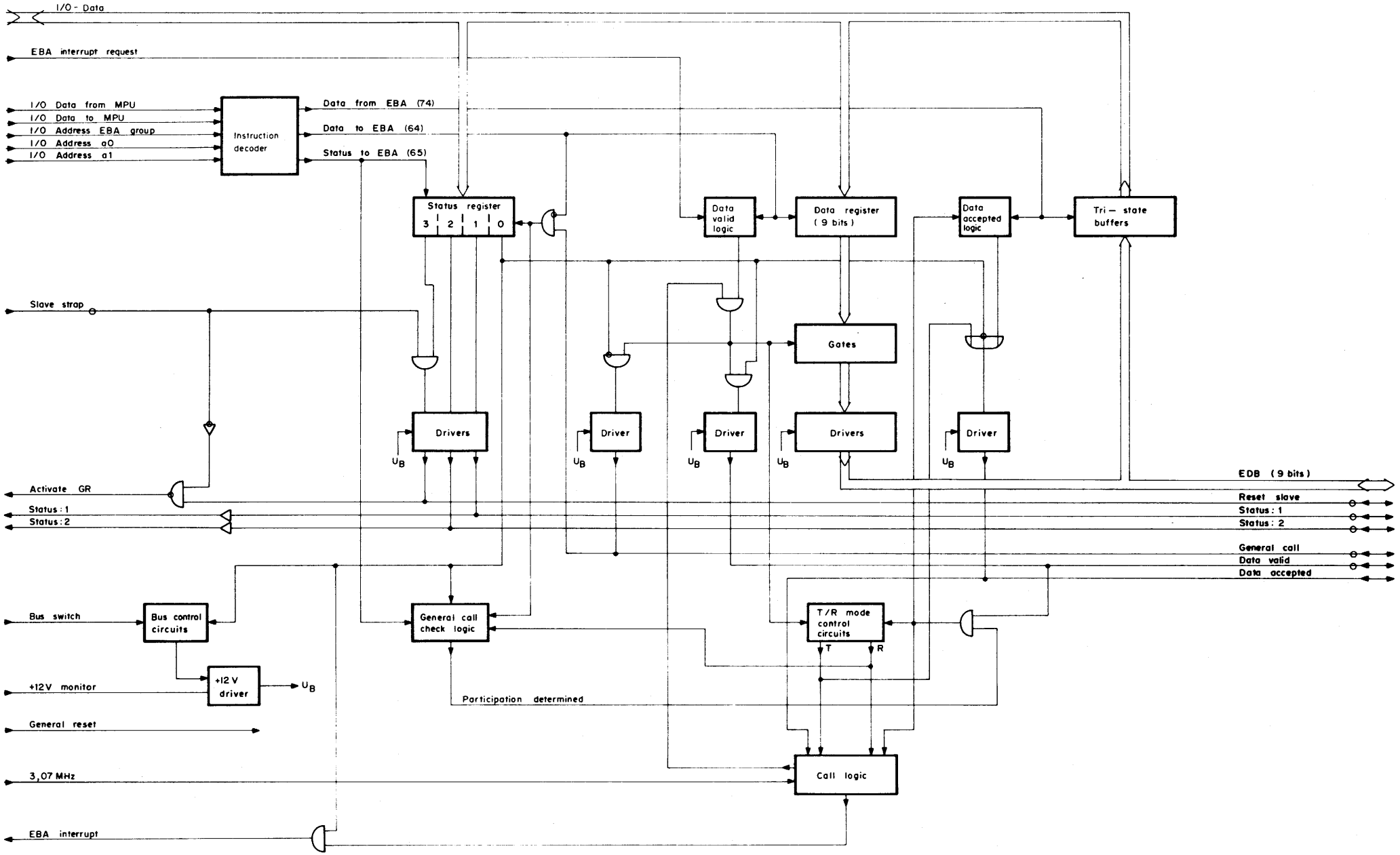


REVISION OF DIAGRAMS AND ASSEMBLY DRAWING

DIAGRAM AND DRAWING NUMBER	SH.	STATUS MOD. / REV.	1
E 07150 2070 3	1	REV.	
— " — 2	2	"	
— " — 1	3	"	
		"	
		"	
		"	
		"	
		"	
		"	
E 07150 0070 3		"	1

INGÅR ÄVEN I / ALSO USED ON				REG.-NR / REG. No	
FÖRST ANVÄND I / FIRST USED ON					
KONTR / CHECK	9603 JB	GODK / APPROV	0266 LB	EBA EXTERNAL BUS ADAPTER	STANSAAB
SKALA / SCALE		KOPIER / COPIED	0673 L-D		E 07150 2070 3 — 3 sh — 1
UTFÖRD / ISSUED	9603 JB	731112			

This document must not be copied, reprinted or disclosed to any third party, either wholly or in part, without our consent. Any infringement will lead to legal proceedings.
STANSAAB elektronik ab SWEDEN



REG NR / REG NO			
KONTR / CHECK	JB	GODK / APPROV	
SKALA / SCALE		KOPIER / COPIED	8506 G.S
UTFÖRD / ISSUED	JB	73 11 12	

EBA
Ext. Bus Adapter

STANSAAB

E 07150 2070 2 | 3 Sh 2

REV. 11

EBA

FÖRST ANVÄND I / FIRST USED ON

INGÅR ÄVEN I / ALSO USED IN

EMJAM (ARBETSMODELL)

kopiera mångfaldigas eller eljest obehörigen utnyttjas. Överlåtelse behöver medgivande STANSAAB Elektronik AB
STANSAAB Elektronik AB
OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!
any third party, either wholly or in part, without our consent.
STANSAAB Elektronik AB SWEDEN

MPU I/O address a1 60
MPU I/O address a0 57
MPU I/O data from MPU 61
MPU I/O address EBA group 22
MPU I/O data to MPU 59

Instruction decoder

74155

Data from EBA (74)
Data to EBA (64)
Status to EBA (65)

EDB Data valid 34

General call check logic

Participation defermined

MPU EBA interr. request 69

Data valid FF

Transmit / receive FF

Receive mode

Call logic

Transmit mode

Data accept FF

MPU 3,07 MHz 20

MPU EBA interrupt 21

EBA reset

Status : 0

Data register

74174

6-BIT

+5V

R69

Status register

74175

4-BIT

1/2-74175

MPU General reset 18

Bus switch 38

MPU +12 V monitor 66

Lamp Bus on 17

Switch Slave strap 68

Bus control circuits

+12V

Z2

Y16

Y17

UB

Y18

Y19

-12V

Activate GR 70

C Status:1 71

C Status:2 72

General call received

7	I/O data:0		MPU
8	" :1		MPU
9	" :2		MPU
10	" :3		MPU
11	" :4		MPU
12	" :5		MPU
13	" :6		MPU
14	" :7		MPU
15	" :80		MPU
16	" :81		MPU
25	EDB : 0		
26	1		
27	2		
28	3		
29	4		
30	5		
31	6		
32	7		
33	8		
35	Data accepted		EDB
36	General call		EDB
1	23	Status:1	EDB
2	24	Status:2	EDB
3	37	Reset slave	EDB

This document must not be copied, reproduced or disclosed to any third party, either wholly or in part, without our written consent. Any infringement will be prosecuted. STANSAAB elektronik AB SWEDEN.

Denna handling får ej utan vårt medgivande kopieras, lånas ut eller delges annan i hela eller delar. Överträdelse beivras med stöd av gällande lag. STANSAAB elektronik AB SVERIGE.

Rev					
INGÅR ÄVEN I ALSO USED ON				REG.-NR REG. No	
FÖRST ANVÄND I FIRST USED ON					
KONTR CHECK	J/S		GODK APPROV		
SKALA SCALE			KOPIEN COPIED	R839 BAR	
UTGÅND ISSUED	9803	7311 12			

EBA
EXTERNAL BUS ADAPTER

STANSAAB
E07150 2070 1 3:h 3

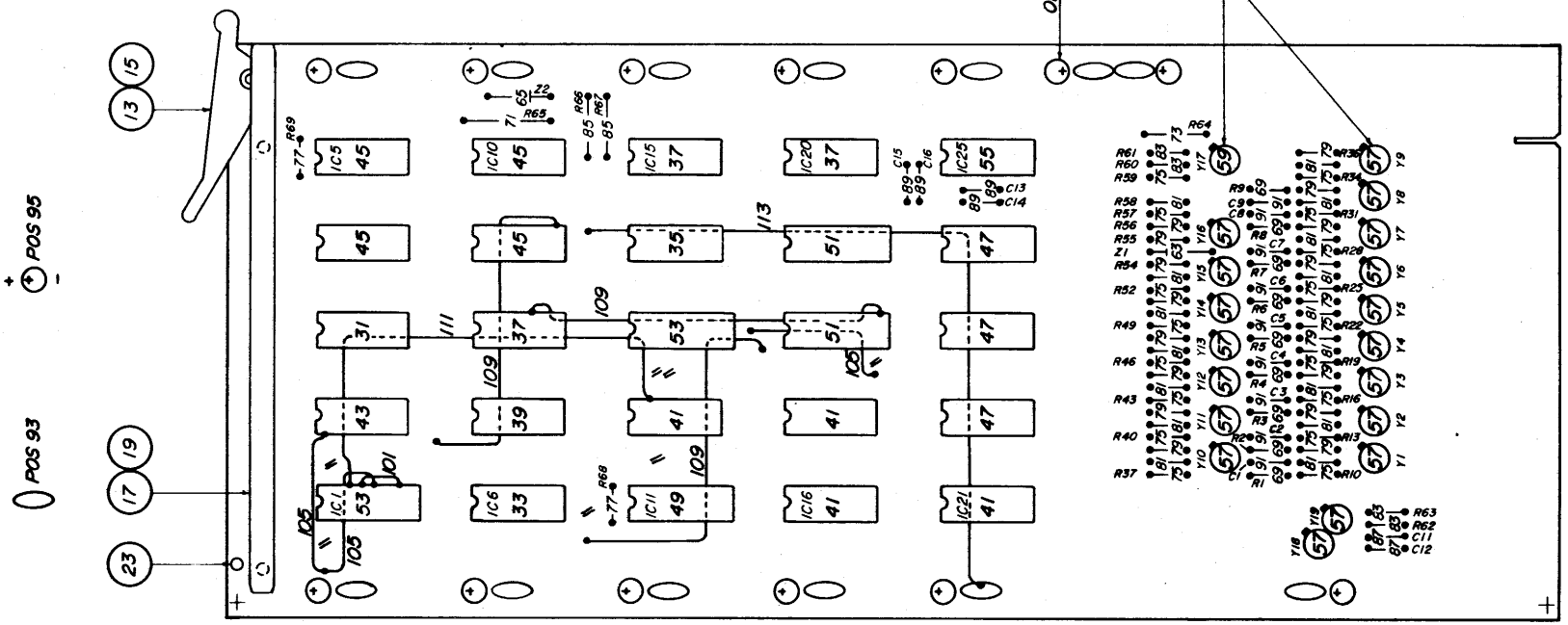
Denna handling får ej, utan vårt medgivande bekantgöras kopieras mångfaldigas eller eljest obehörigen utnyttjas. Överträdelse beivras med stöd av gällande lag.
STANSAAB Elektronik AB

OM NAGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!

This document must not be copied, reprinted or disclosed to any third party, either wholly or in part, without our consent. Any infringement will lead to legal proceedings.
STANSAAB Elektronik AB SWEDEN

EBA

REV. 1 740509 | E07150 0070 3



// = FÖRBINDNINGEN BRYTES 7 ST

TILLÅTEN LÖDHÖJD 3MM

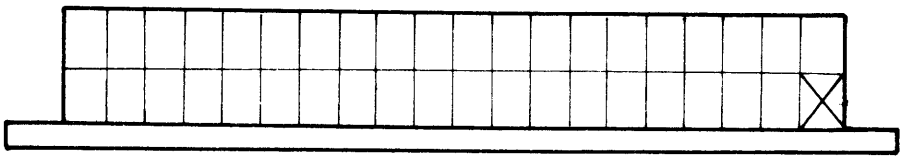
REG NR / REG NO			EBA
KONTR / CHECK	dB.	GODK / APPROV 0266 LB	
SKALA / SCALE	1:1	KOPIER / COPIED 8327 KGB	KRETSKORT / PRINTED BOARD ASSEMBLY
UTFÖRD / ISSUED 9603 JB		73 11 30	E07150 0070 3

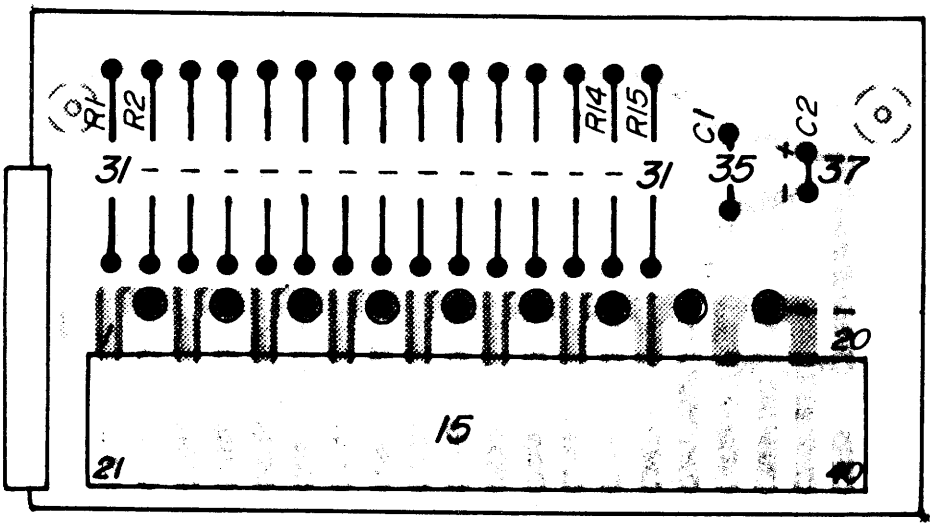
STANSAAB

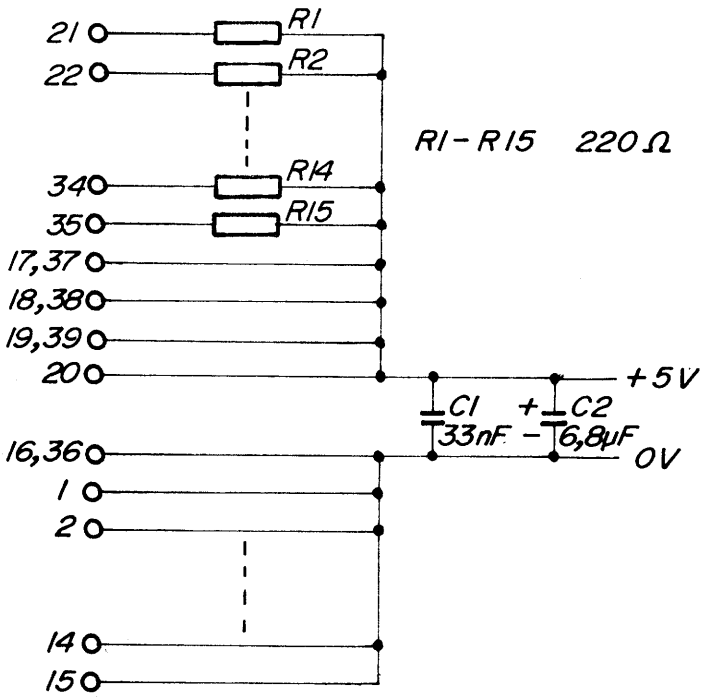
FÖRST ANVAND I / FIRST USED ON

INGÅR ÄVEN I / ALSO USED ON

E 28012 0000 4







REV. 1				BUSSAVSLUTNING BUS TERMINATOR KRETSKORT PRINTED BOARD ASSEMBLY
	CHECK		APPROV.	0266 LB
	SKALA SCALE	2:1	KOPIER COPIED	9696 HP
	UTFÖRD ISSUED	0878 DA		731120

STANSAAB

E28012 0000 4

DRAWINGS AND DIAGRAMS

Display Unit 3510

Part 7	Cable Run Sheets		
		DU 3510 – KB 3540	E27403 4000 401
		ID – KB 3540	E27411 4000 401
		DU 3510 – Power Supply – Frame	One sheet
	CRT Unit		
		Circuit diagram	66 83841–01A
		Power supply	66 80154–01
		Pulse diagram	66 83841–01
	STL, Serial Transfer Logic		
		Reference sheet	E27409 2001 3, sh 1
		Block diagram	E27409 2001 3, sh 2
		Logic diagram	E27409 2001 1, sh 3–4
		PCB assembly	E27409 0001 2, sh 1
Part 8	ID, Identification Device		
		Reference sheet	E27411 2000 3, sh 1
		Block diagram	E27411 2000 3, sh 2
		Logic diagram	E27411 2000 7, sh 3
		PCB assembly	E24417 0000 3
	KB, Keyboard		
		Logic diagram	E00101 2000 1
		PCB assembly	E00101 0000 2

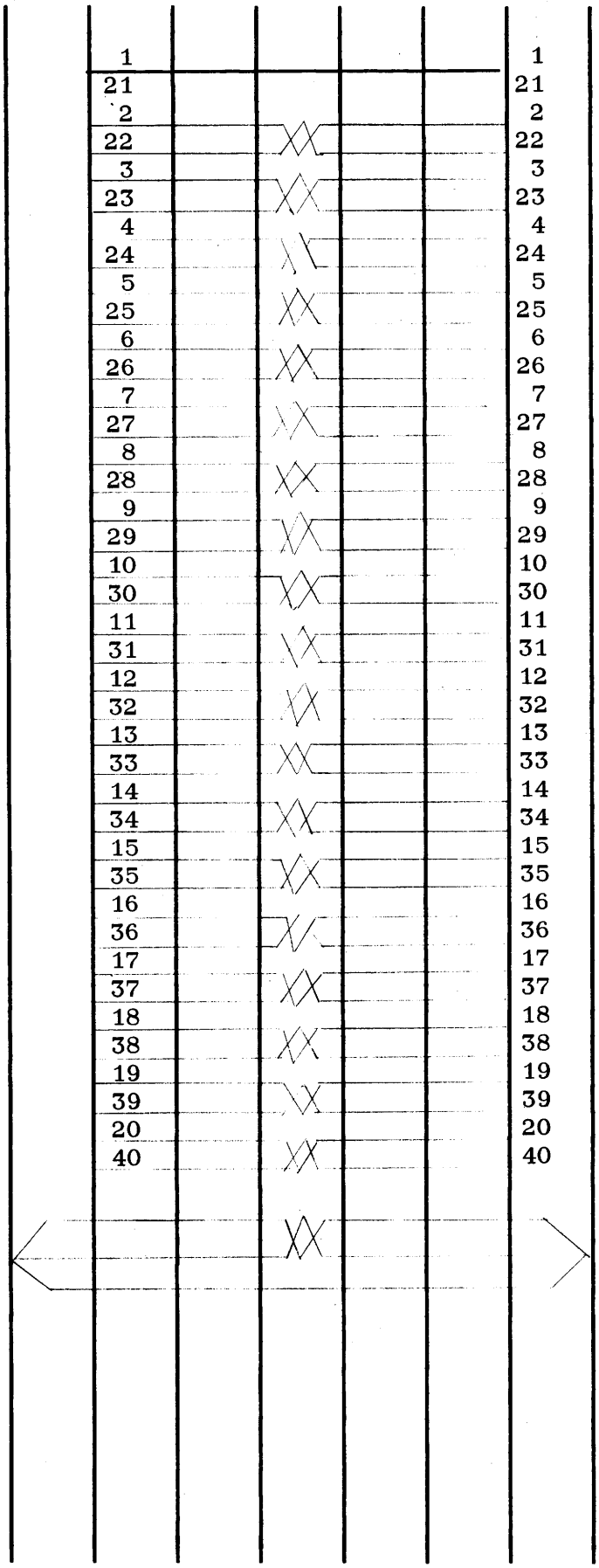
Display Unit

STL

Chassis P3

Keyboard

P1 Chassis

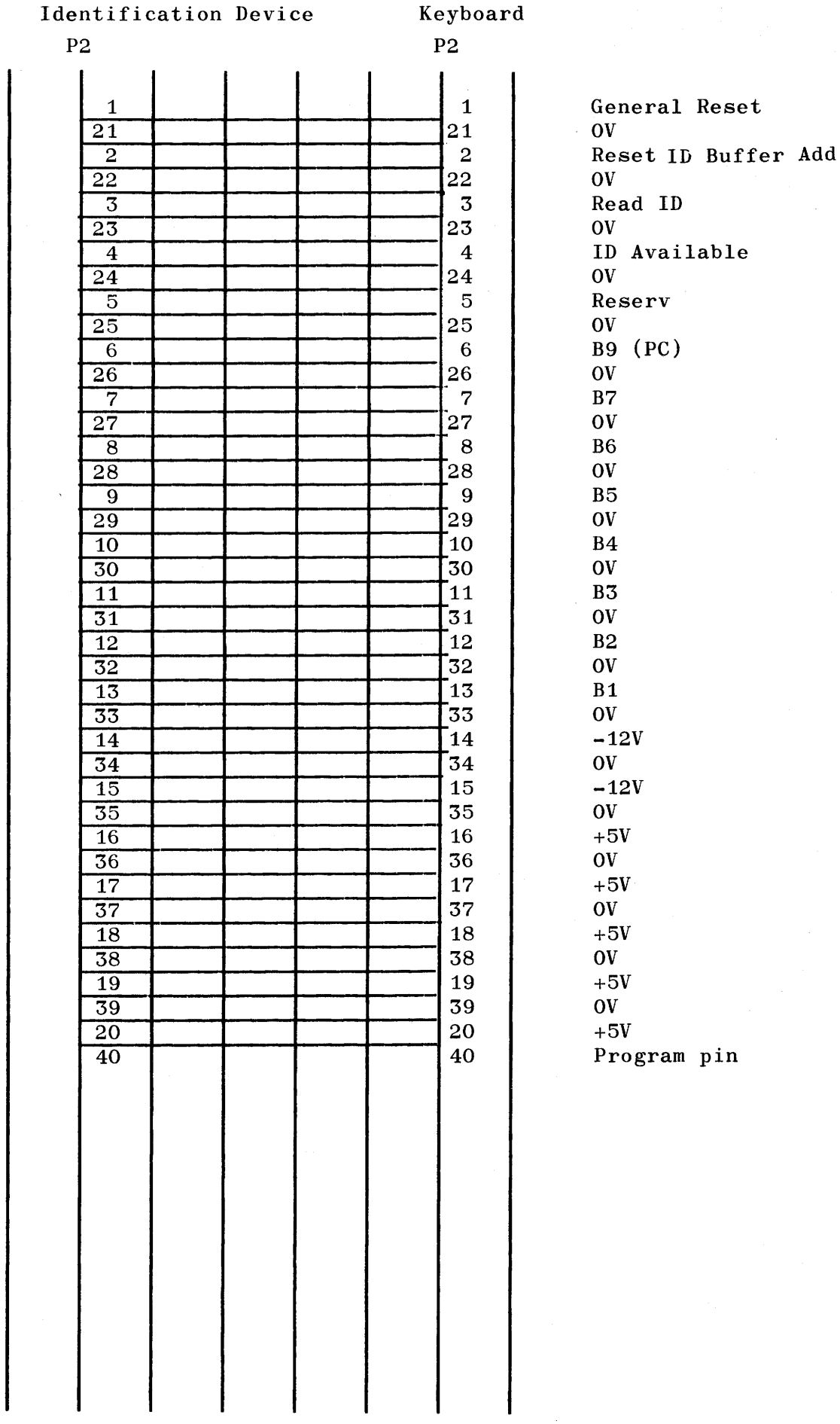


Chassis P3	P1 Chassis	
1	1	+5V
21	21	Program pin
2	2	+5V
22	22	OV
3	3	+5V
23	23	OV
4	4	-12V
24	24	OV
5	5	-12V
25	25	OV
6	6	B1
26	26	Repeat
7	7	B2
27	27	B3
8	8	B4
28	28	B5
9	9	OV
29	29	B6
10	10	B7
30	30	B8
11	11	B9 (PC)
31	31	OV
12	12	ID Available
32	32	OV
13	13	Read ID
33	33	OV
14	14	Reset ID Buff Addr.
34	34	General Reset
15	15	Strobe 1 Data Avail.
35	35	OV
16	16	Read KB
36	36	OV
17	17	Step KB
37	37	OV
18	18	Click
38	38	OV
19	19	IM ⊗ I/M-lamp
39	39	IM ⊗ - " -
20	20	USM ⊗ USM-lamp
40	40	USM ⊗ - " -
		Protective ground

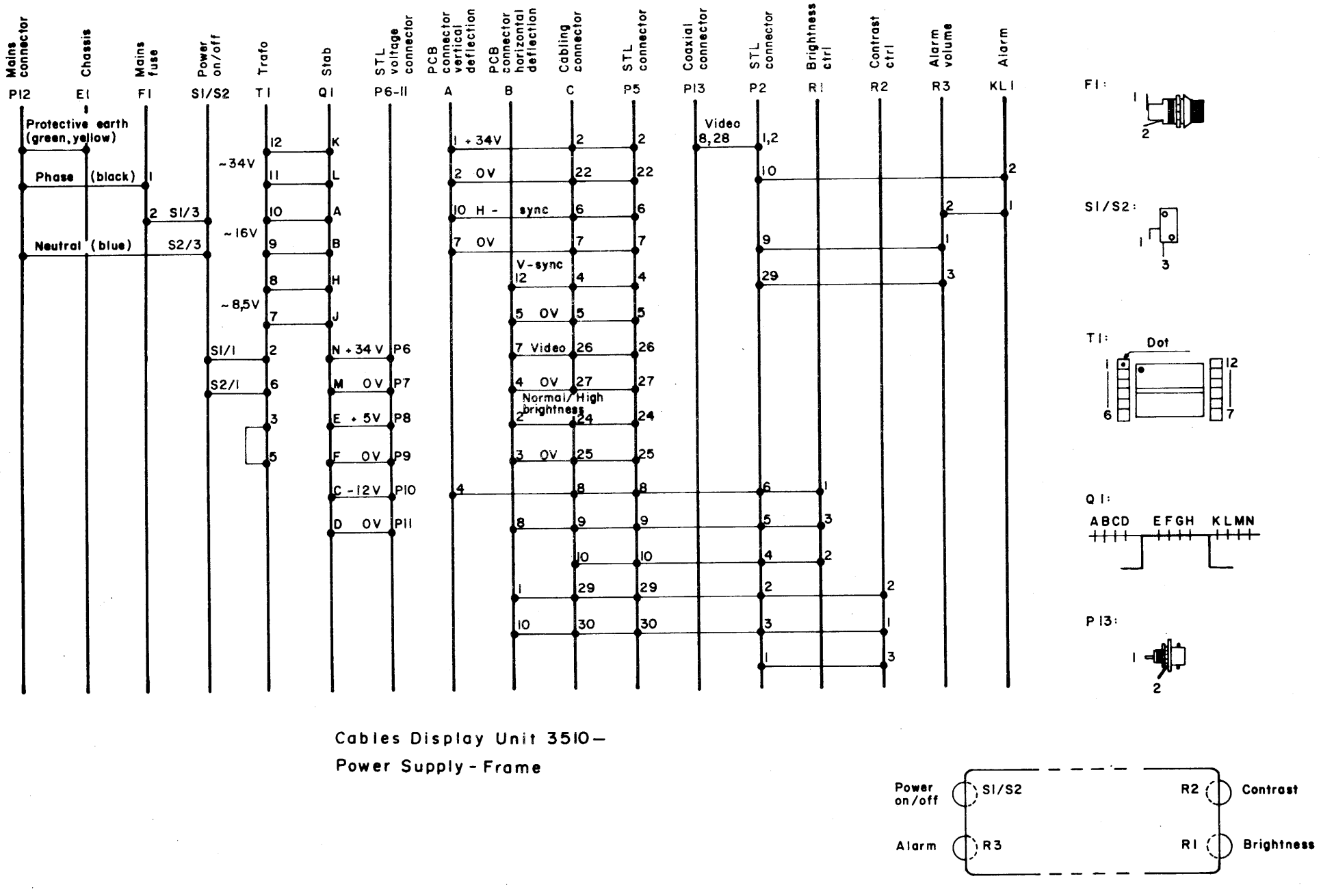
CABLE DISPLAY UNIT 3510 -

- KEYBOARD 3540

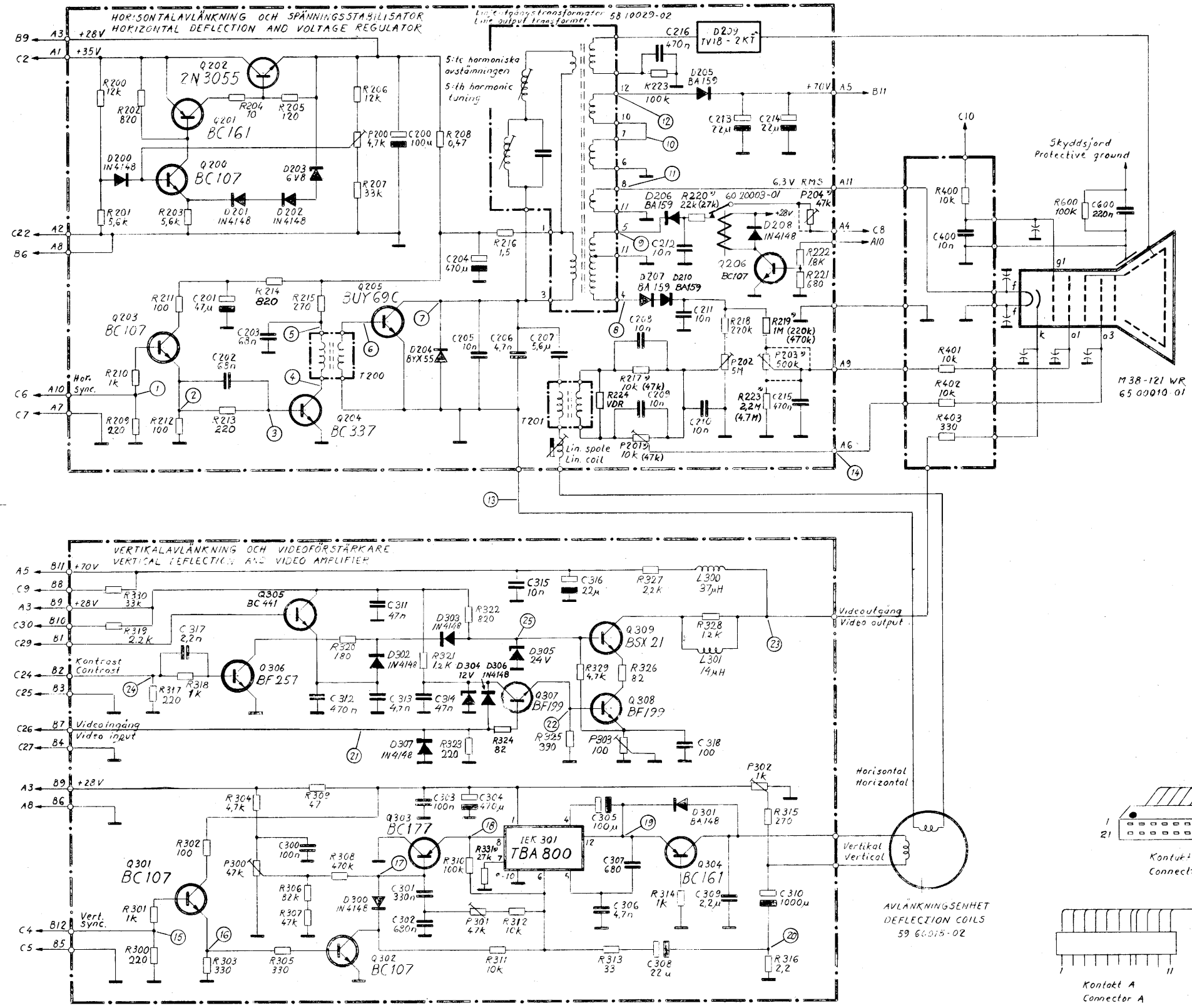
E27403 4000 401



CABLE ID - KEYBOARD E27411 4000 401



Cables Display Unit 3510–
Power Supply - Frame



(C) Kondensatorer · Capacitors

Nr No	Art.nr Part no	Värde Value	Tol.± %	Mat	Max V	Pos Item
200	62 21308-01	100 µF	-10 +50	Elkyt	40	A1
201	62 21305-01	47 µF	-10 +50	"	25	A1
202	62 21293-01	68 nF	20	Met	400	A1
203	62 21348-01	68 nF	10	"	250	A1
204	62 20036-01	470 µF	-10 +50	Elkyt	40	A2
205	62 20034-01	10 nF	5	Polycarb.	1600	A2
206	62 20010-01	47 nF	5	"	400	A2
207	62 12701-01	5,6 µF	10	Polyest.	100	A2
208	62 21095-01	10 nF	20	Popper	1000	A2
209	62 21095-01	10 nF	20	"	1000	A2
210	62 21095-01	10 nF	20	"	1000	A2
211	62 21095-01	10 nF	20	"	1000	A2
212	62 21095-01	10 nF	20	"	1000	A2
213	62 21311-01	22 µF	-20+100	Elkyt	100	A2
214	62 21311-01	22 µF	-20+100	"	100	A2
215	62 20060-01	470 nF	20	Polyest.	630	A2
216	62 20045-01	470 nF	5	Polycarb.	100	A2
300	62 21023-01	100 nF	10	Polyest.	100	B1
301	62 21309-01	330 nF	10	"	250	B1
302	62 21333-01	680 nF	10	"	100	B1
303	62 21023-01	100 nF	10	"	100	B1
304	62 20036-01	470 µF	-10 +50	Elkyt	40	A2
305	62 21175-01	100 µF	-20 +50	"	25	B2
306	62 21338-01	47 nF	20	Keram	500	B2
307	62 20009-01	680 pF	2,5	Styrol	63	B2
308	62 50024-01	22 µF	5	Tantal	10	B2
309	62 21073-01	2,2 µF	10	Polyest.	100	B2
310	62 21353-01	1000 µF	-20 +50	Elkyt	25	B2
311	62 21186-01	47 nF	10	Polyest.	250	B1
312	62 21048-01	470 nF	20	"	100	B1
313	62 21338-01	47 nF	20	Keram	500	B1
314	62 21186-01	47 nF	10	Polyest.	250	B1
315	62 21288-01	10 nF	20	"	400	B2
316	62 21311-01	22 µF	-20 +100	Elkyt	100	B2
317	62 30010-01	22 nF	5	Styrol	160	B2
318	62 21254-01	100 pF	5	Keram	500	B2
319						
400	62 21288-01	10 nF	20	Polyest.	400	A3
CCC1	62 21297-01	220 nF	20	Polyest.	100	A3

(P) Potentiometrar · Potentiometers

Nr No	Art.nr Part no	Värde Value	Funktion	Pos Item
200	61 17960-01	4,7 kΩ	Horizontal gain (bredd)	A1
201	61 17952-01	10 kΩ	Dynamic focus-voltage	A2
202	61 17699-01	5 MΩ	Static focus-voltage	A2
203	61 17642-01	500 kΩ	Screen-grid voltage	A2
204	61 17954-01	47 kΩ	Black level	A2
300	61 17705-01	47 kΩ	Vert. gain (höjd)	B1
301	61 17705-01	47 kΩ	" linearity	B2
302	61 17949-01	1 kΩ	" positioning	B2
303	61 17700-01	100 Ω	Video-amp compensation	B2

(R) Motstånd · Resistors

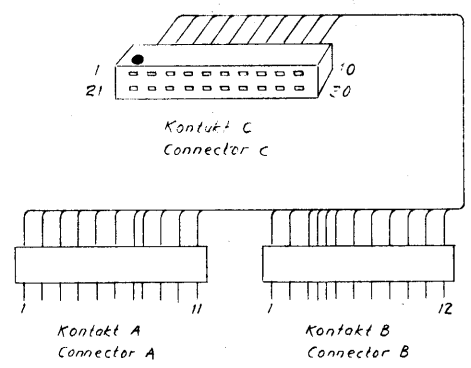
Nr No	Art.nr Part no	Värde Value	Tol.± %	Mat.	Max W	Pos Item
200	61 29776-01	12 kΩ	5	Carbon film	0,5	A1
201	61 29722-01	5,6 kΩ	5	"	0,5	A1
202	61 29706-01	820 Ω	5	"	0,5	A1
203	61 29722-01	5,6 kΩ	5	"	0,5	A1
204	61 29685-01	10 Ω	5	"	0,5	A1
205	61 29607-01	120 Ω	5	"	0,5	A1
206	61 29726-01	12 kΩ	5	"	0,5	A1
207	61 29731-01	33 kΩ	5	"	0,5	A1
208	61 29619-01	0,47 Ω	10	Wirewound	4	A1
209	61 29710-01	220 Ω	5	Carbon film	0,5	A1
210	61 29707-01	1 kΩ	5	"	0,5	A1
211	61 29378-01	100 Ω	5	"	0,5	A1
212	61 29378-01	100 Ω	5	"	0,5	A1
213	61 29700-01	220 Ω	5	"	0,5	A1
214	61 29706-01	820 Ω	5	"	0,5	A1
215	61 10024-01	270 Ω	5	Wirewound	4	A1
216	61 29822-01	1,5 Ω	10	"	4	A2
217	61 29379-01	10 kΩ	5	Carbon film	0,5	A2
218	61 29741-01	220 kΩ	5	"	0,5	A2
219	61 20367-01	4,7 MΩ	10	Composition	0,5	A2
220	61 29730-01	27 kΩ	5	Carbon film	0,5	A2
221	61 29489-01	680 Ω	5	"	0,5	A2
222	61 29710-01	18 kΩ	5	"	0,5	A2
300	61 29730-01	220 Ω	5	Carbon film	0,5	B1
301	61 29707-01	1 kΩ	5	"	0,5	B1
302	61 29378-01	100 Ω	5	"	0,5	B1
303	61 29378-01	100 Ω	5	"	0,5	B1
304	61 29721-01	4,7 kΩ	5	"	0,5	B1
305	61 29702-01	330 Ω	5	"	0,5	B1
306	61 29736-01	82 kΩ	5	"	0,5	B1
307	61 29521-01	47 kΩ	20	NTC	0,6	B1
308	61 29745-01	470 kΩ	5	Carbon film	0,5	B1
309	61 29828-01	47 Ω	10	Wirewound	4	B1
310	61 29737-01	100 kΩ	5	Carbon film	0,5	B2
311	61 29725-01	10 kΩ	5	"	0,5	B2
312	61 29725-01	10 kΩ	5	"	0,5	B2
313	61 29631-01	33 Ω	5	"	0,5	B2
314	61 29707-01	1 kΩ	5	"	0,5	B2
315	61 29701-01	270 Ω	5	"	0,5	B2
316	61 29681-01	2,2 Ω	5	"	0,5	B2
317	61 29707-01	220 Ω	5	"	0,5	B1
318	61 29707-01	1 kΩ	5	"	0,5	B1
319	61 29711-01	22 kΩ	5	"	0,5	B1
320	61 29699-01	180 Ω	5	"	0,5	B1
321	61 29708-01	1,2 kΩ	5	"	0,5	B1
322	61 29706-01	820 Ω	5	"	0,5	B1
323	61 29700-01	220 Ω	5	"	0,5	B1
324	61 29549-01	82 Ω	5	"	0,5	B2
325	61 29703-01	390 Ω	5	"	0,5	B2
326	61 29696-01	82 Ω	5	"	0,5	B2
327	61 29711-01	2,2 kΩ	5	"	0,5	B2
328	61 29708-01	1,2 kΩ	5	"	0,5	B2
329	61 29263-01	47 kΩ	5	"	0,3	B2
330	61 29731-01	33 kΩ	5	Carbon film	0,5	B1
223	61 20369-01	2,8 MΩ	5	"	0,5	A2
331	61 29730-01	27 kΩ	5	"	0,5	B1
400	61 29277-01	10 kΩ	10	Composition	0,5	A3
401	61 29277-01	10 kΩ	10	"	0,5	A3
402	61 29277-01	10 kΩ	10	"	0,5	A3
403	61 29605-01	330 Ω	5	"	0,5	A3
R600	61 00010-01	100 kΩ	⌀ 2	Composition	0,35	A3

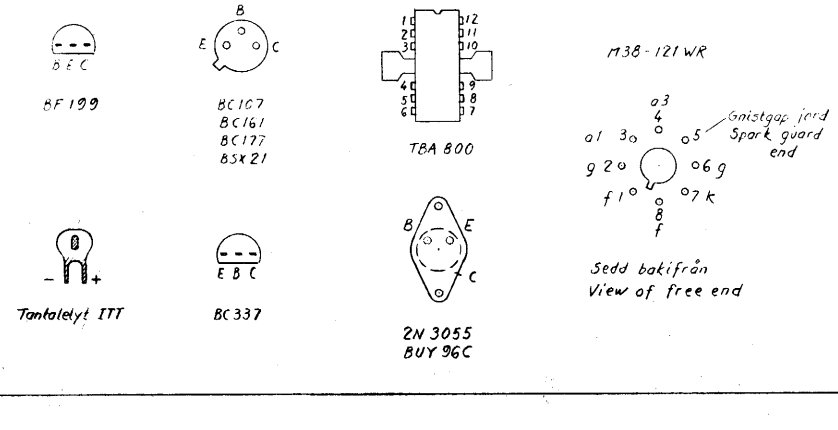
(D) Dioder · Diodes

Nr No	Art.nr Part no	Typ Type	Pos Item
200	63 08824-01	IN4148	A1
201	63 08824-01	IN4148	A1
202	63 08824-01	IN4148	A1
203	63 40025-01	6V8 zener/0,4W	A1
204	63 40023-01	BYX 55	A1
205	63 08751-01	BA159	A2
206	63 08751-01	BA159	A2
207	63 08751-01	BA159	A2
208	63 08824-01	IN4148	A2
209	63 08917-01	TV18-2KT	A2
300	63 08824-01	IN4148	B1
301	63 08824-01	BA148	B2
302	63 08824-01	IN4148	B1
303	63 08824-01	IN4148	B1
304	63 40034-01	12V zener/0,4W	B2
305	63 08721-01	24V zener/0,4W	B2
306	63 08824-01	IN4148	B2
307	63 08824-01	IN4148	B1
210	63 08751-01	BA159	A2

(Q) Transistorer · Transistors

Nr No	Art.nr Part no	Typ Type	Pos Item
200	63 09087-01	BC107	A1
201	63 09221-01	BC161	A1
202	63 20014-01	2N3055	A1
203	63 09087-01	BC107	A1
204	63 10017-01	BC337	A1
205	63 20022-01	BUY69C	A1
301	63 09087-01	BC107	B1
302	63 09087-01	BC107	B1
303	63 09135-01	BC177	B1
304	63 09221-01	BC161	B2
305	63 10018-01	BC441	B1
306	63 00015-01	BF	B1
307	63 00005-02	BF199	B2
308	63 00005-02	BF199	B2
309	63 20019-01	BSX21	B2





(L) Spolar · Coils

Nr No	Art.nr Part no	Benämning	Pos Item
300	58 56779-01	37 µH	B2
301	58 00022-01	14 µH	B2

(T) Transformatorer · Transformers

Nr No	Art.nr Part no	Typ Type	Pos Item
200	58 10030-01	Drive transformer	A1
201	58 10033-01	Focus-transformer	A2

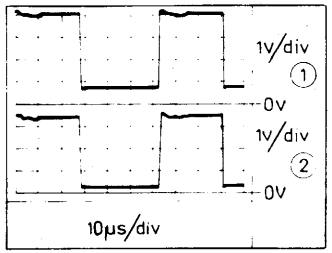
(IEK) Integrerade elektroniska kretsar · Integrated circuits

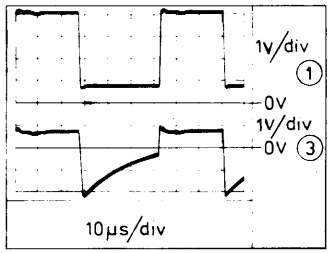
Nr No	Art.nr Part no	Typ Type	Pos Item
301	63 10002-01	TBA 800	B2

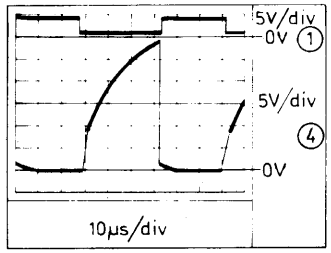
x) Modifikationslapp märkt · Modifications-tag marked

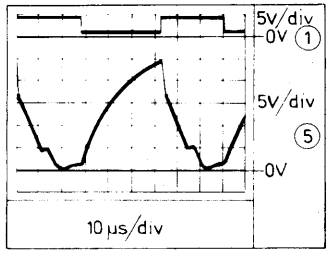
R217 = 47k } P201 = 47k
R220 = 27k } P204 = shorted
R219 = 220k } P203 = 500k } R223 = 4,7 M
R219 = 1M } P203 = shorted } R223 = 2,2 M
R331 = 27k
R219=470k P203=500k R223=2,2M
R331 = ∞

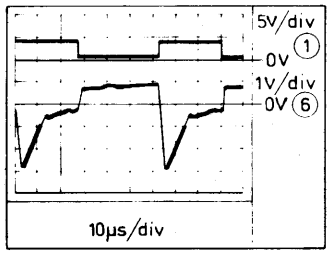
Benämning	KOPPLINGSSCHEMA BILDENHET · CIRCUIT DIAGRAM MONITOR		
	17 33941-01/02		
Ritad	74-02-25	E	
Kop.		Format A1	A4 1 Ark 2
Granskad	74-09-11		
Godk	74-09-11	66 03841-01A	
Reg	74-09-11		

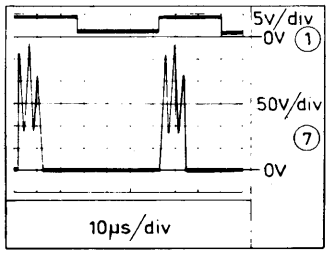


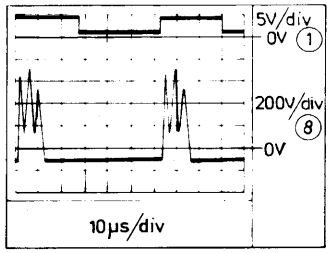


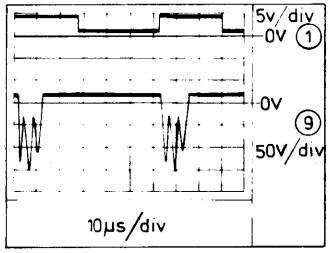


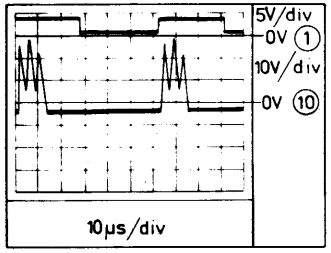


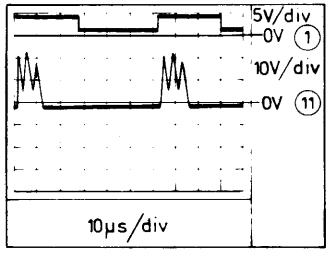


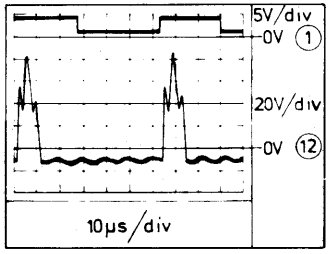


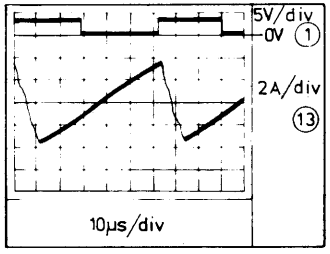


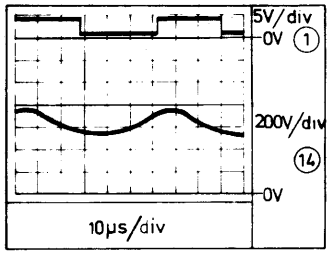


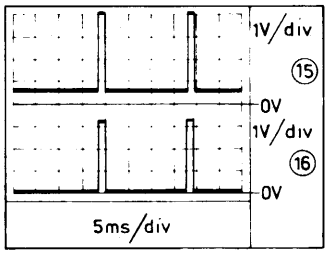


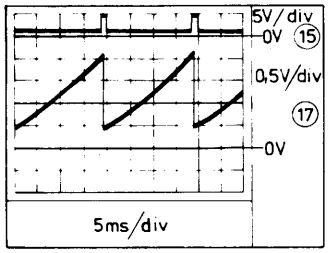


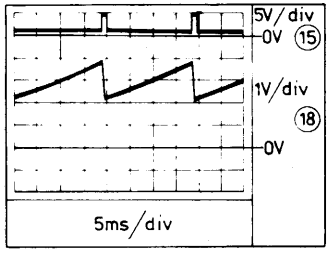


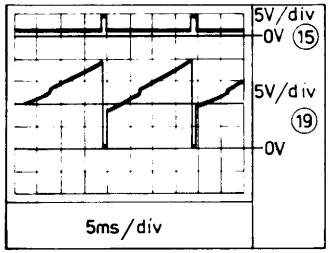


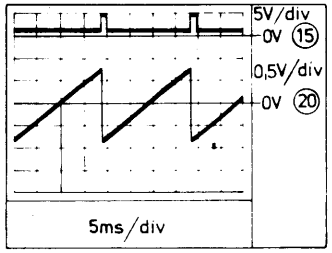


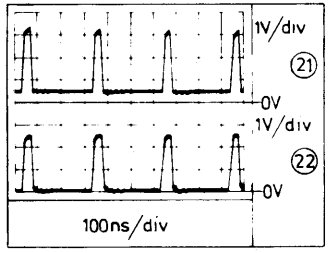


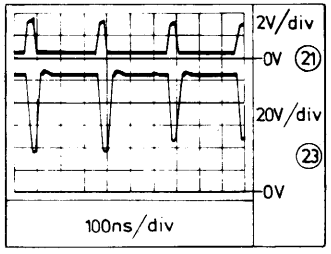


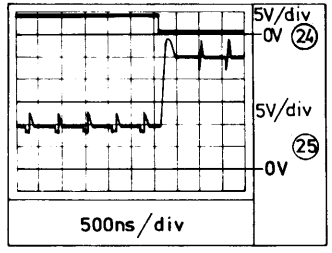








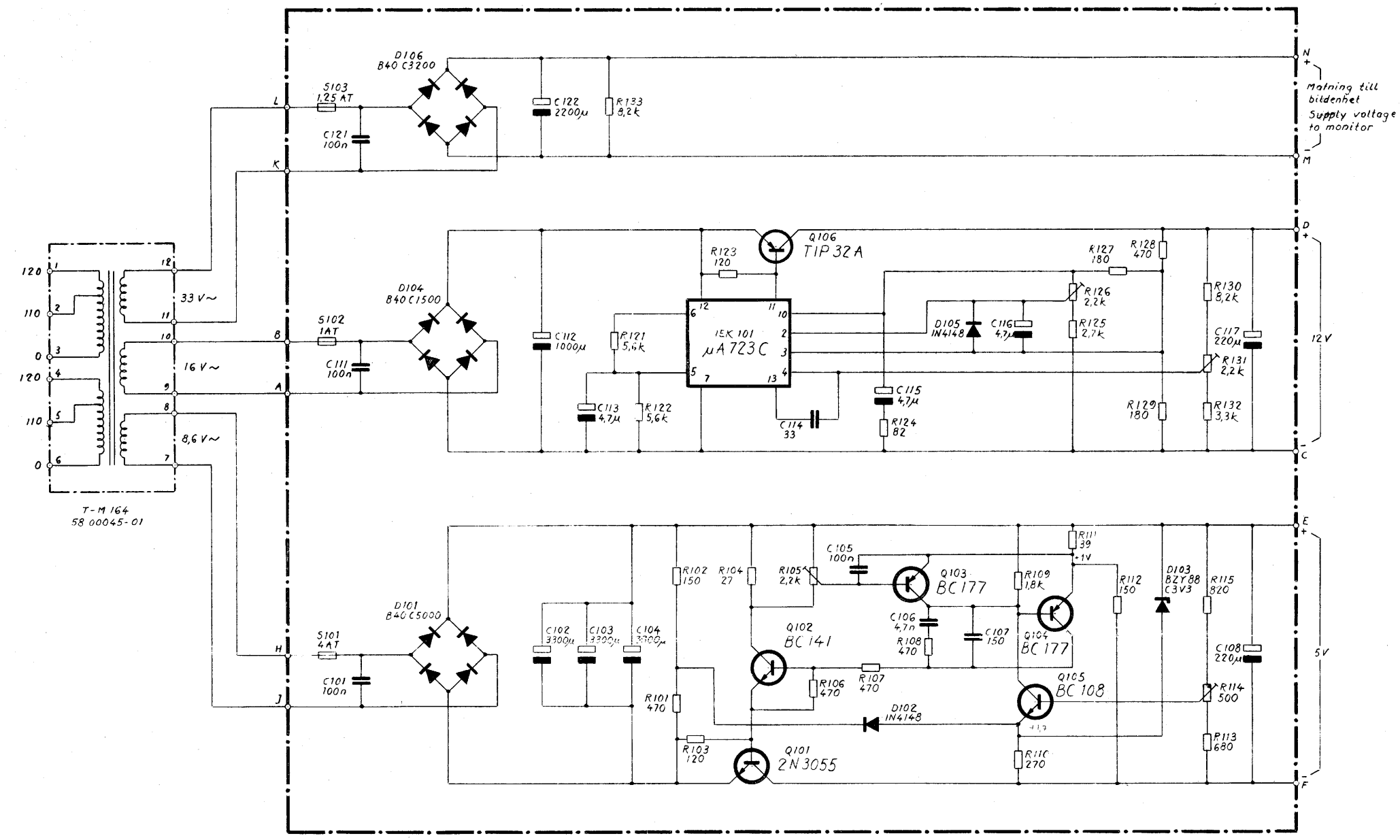


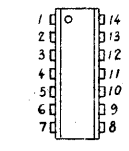


ALL MEASUREMENTS ARE MADE WITH A 50 MHz DC-COUPLED OSCILLOSCOPE USING A 10:1 PROBE EXCEPT FOR TESTPOINT 23 WHERE A LOW CAPACITIVE 100:1 PROBE SHOULD BE USED TO AVOID SERIOUS LOADING OF THE VIDEO AMPLIFIER. FOR TESTPOINTS 21 - 25 THE GRAPHS SHOW RESPONSE WHEN THE CHARACTER: IS GENERATED BY THE STANSAAB TESTPATTERN GENERATOR. IN TESTPOINT 23 THE TWO FIRST PULSES SHOW VIDEO AMPLIFIER RESPONSE WHEN "CONTRAST" CONTROL INPUT IS HIGH AND THE TWO LATTER WHEN THIS INPUT IS LOW, I.E. NORMAL INTENSITY.

ALLA MÄTNINGAR HAR UTFÖRTS MED ETT 50 MHz DC-KOPPLAT OSCILLOSKOP. PROBARNA BÖR VARA 10:1 UTOM VID MÄTNING I TESTPUNKT 23 DÄR EN LÅG-KAPACITIV PROB (100:1 T.EX.) BÖR ANVÄNDAS FÖR ATT UNDVIKA ALLTFÖR STOR BELASTNING AV VIDEOFÖRSTÄRKAREN. I MÄTPUNKTERNA 21 - 25 VISAR BILDERNA SPÄNNINGAR NÄR TECKNET: GENERERAS AV STYRENHETEN. I PUNKT 23 VISAR DE TVÅ FÖRSTA PULSERNA VIDEOFÖRSTÄRKARENS UTGÅNG NÄR "CONTRAST" STYRPULSEN ÄR HÖG OCH DE TVÅ SENARE NÄR DENNA PULS ÄR LÅG, DVS. FÖR NORMAL LJUSSTYRKA.

BENÄMNING	OSCILLOSKOPBILDER OSCILLOGRAPHS FOR MONITOR 17 33841 01/02				17049
RITAD					
KOP.	74.05.28		FORMAT A2	BLAD 2 AV 2	
GRANSK.	74.09.11				
GODK.	74-09-11		66 83841-01		
REG.	74-09-11				
LUXOR INDUSTRI AKTIEBOLAG MOTALA					

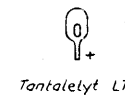




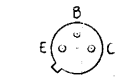
µA 723C



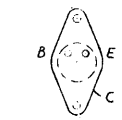
Tantaleiyt ITT



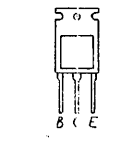
Tantaletyt LTT



BC 108
BC 141
BC 177



ZN 3055



TIP 32A

(R) Motstånd Resistors

Nr No	Art.nr Part no	Värde Value	Tol. ± %	Mat	Max. W	Pos. Item
101	61 29704-01	470 Ω	5	Kolskikt	0,5	C2
102	61 29698-01	150 Ω	5	-"-	0,5	C2
103	61 29697-01	120 Ω	5	-"-	0,5	C2
104	61 29690-01	27 Ω	5	-"-	0,5	C2
105	61 17703-01	2,2 kΩ	20	Trimpot	0,1	C2
106	61 29704-01	470 Ω	5	Kolskikt	0,5	C2
107	61 29704-01	470 Ω	5	-"-	0,5	C2
108	61 29704-01	470 Ω	5	-"-	0,5	C2
109	61 29710-01	1,8 kΩ	5	-"-	0,5	C3
110	61 29701-01	270 Ω	5	-"-	0,5	C3
111	61 29692-01	39 Ω	5	-"-	0,5	C3
112	61 29698-01	150 Ω	5	-"-	0,5	C3
113	61 29689-01	680 Ω	5	-"-	0,5	C3
114	61 17549-01	500 Ω	20	Trimpot	0,15	C3
115	61 29706-01	820 Ω	5	Kolskikt	0,5	C3
121	61 29722-01	5,6 kΩ	5	Kolskikt	0,5	B2
122	61 29722-01	5,6 kΩ	5	-"-	0,5	B2
123	61 29697-01	120 Ω	5	-"-	0,5	B2
124	61 29696-01	82 Ω	5	-"-	0,5	B2
125	61 29718-01	2,7 kΩ	5	-"-	0,5	B3
126	61 17703-01	2,2 kΩ	20	Trimpot	0,1	B3
127	61 29699-01	180 Ω	5	Kolskikt	0,5	B3
128	61 29704-01	470 Ω	5	-"-	0,5	B3
129	61 29699-01	180 Ω	5	-"-	0,5	B3
130	61 29724-01	8,2 kΩ	5	-"-	0,5	B3
131	61 17703-01	2,2 kΩ	20	Trimpot	0,1	B3
132	61 29719-01	3,3 kΩ	5	Kolskikt	0,5	B3
133	61 29724-01	8,2 kΩ	5	-"-	0,5	A2

(C) Kondensatorer Capacitors

Nr No	Art.nr Part no	Värde Value	Tol. ± %	Mat	Max. V	Pos. Item
101	62 21023-01	100 nF	10	Polyester	100	C1
102	62 21713-02	3300 µF	-10 +100	Elyt	25	C1
103	62 21713-02	3300 µF	-10 +100	-"-	25	C2
104	62 21713-02	3300 µF	-10 +100	-"-	25	C2
105	62 21023-01	100 nF	10	Polyester	100	C2
106	62 21338-01	4,7 nF	20	Keram.	500	C2
107	62 21256-01	150 pF	5	-"-	500	C3
108	62 21306-01	220 µF	-10 +50	Elyt	25	C3
111	62 21023-01	100 nF	10	Polyester	100	B1
112	62 21353-01	1000 µF	-20 +50	Elyt	25	B1
113	62 50020-01	4,7 µF	-20 +50	Tantal elyt	25	B2
114	62 21243-01	33 pF	2	Keram	63	B2
115	62 50020-01	4,7 µF	-20 +50	Tantal elyt	25	B2
116	62 50020-01	4,7 µF	-20 +50	-"-	25	B3
117	62 21306-01	220 µF	-10 +50	Elyt	25	B3
121	62 21023-01	100 nF	10	Polyester	100	A1
122	62 50001-01	2200 µF	-10 +50	Elyt	63	A1

(Q) Transistorer Transistors

Nr No	Art.nr Part no	Typ Type	Pos Item
101	63 30014-01	2N 3055	C2
102	63 09220-01	BC 141	C2
103	63 09135-01	BC 177	C2
104	63 09135-01	BC 177	C3
105	63 09162-01	BC 108	C3
106	63 20013-01	TIP 32A	B2

(D) Dioder Diodes

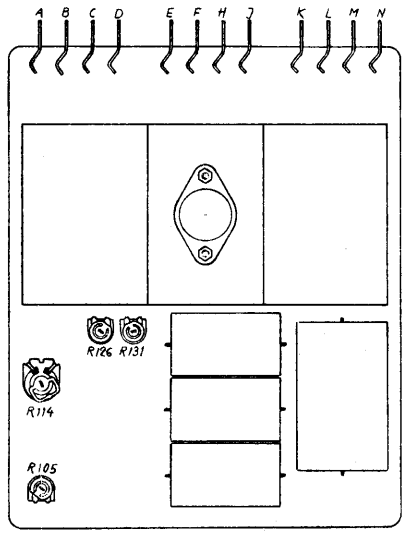
Nr No	Art.nr Part no	Typ Type	Pos Item
101	63 40032-01	B40 C5000	C1
102	63 08824-01	IN 4148	C2
103	63 08746-01	BZY88 C3V3	C3
104	63 40031-01	B40 C1500	B1
105	63 08824-01	IN 4148	B3
106	63 88901-01	B40 C3200	A1

(IEK) Integrerade elektroniska kretsar Integrated circuits

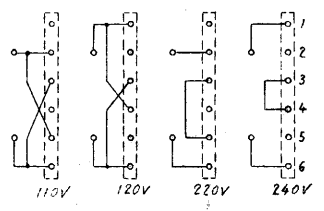
Nr No	Art.nr Part no	Typ Type	Pos. Item
101	64 50000-01	µA 723C	B2

(S) Säkringar Fuses

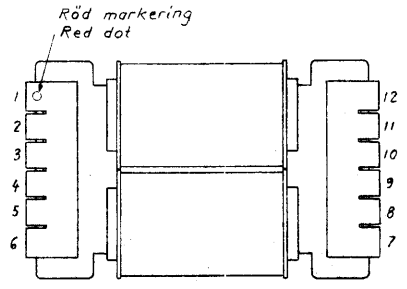
Nr No	Art.nr Part no	Värde Value	Pos Item
101	65 83214-01	4 AT	C1
102	65 82669-01	1 AT	B1
103	65 83007-01	1,25 AT	A1



Koppling vid olika nätspänningar
Wiring for different line voltages



Röd markering
Red dot



T-M164
58 00045-01

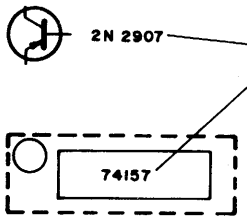
Benämning
KOPPLINGSSCHEMA KRAFTENHET
CIRCUIT DIAGRAM POWER SUPPLY
ES 30154-01

Ritad	76-01-25	C	1
Kop.		FORMAT A1	BLAD av.
Granskad		366 80154-01	
Godk.			
Reg.			

LUXOR INDUSTRI AKTIEBOLAG MOTALA

Denna handling får ej utan vårt medgivande bekantgöras, kopieras, mångfaldigas eller eljest obehörigen utnyttjas. Överträdelse bevras med stöd av gällande lag.
STANSAAB elektronik ab

TEST POINTS	
NR	SIGNAL
1	NORM/HIGH BRIGHTNESS TO DISPLAY
2	H-SYNC TO DISPLAY
3	V-SYNC TO DISPLAY
4	VIDEO TO DISPLAY
5	CP TO BYTE COUNTER
6	RESET BYTE COUNTER
7	BYTE 0-3
8	RESET BIT COUNTER
9	CP TO BIT COUNTER
10	SET V-SYNC GENERATOR
11	SET H-SYNC GENERATOR
12	$\overline{\text{H-SYNC}}$
13	$\overline{\text{Q}}$ PC-GENERATOR
14	OUTPUT FROM KB-ID-LP/DATA REG
15	PROGRAM START PULS
16	OUTPUT FROM TAG REGISTER
17	CP TO REC REGISTER CMD
18	SET PULS CMD REGISTER
19	SET ID IN/OUT - BYTE
20	SET-PULS KB-ID-LP/DATA REG
21	STROBE TO TAG REGISTER
22	CP TO TAG REGISTER
23	DATA/ VIDEO IN/OUT
24	-5V
25	-4V
26	0V
27	
28	
29	
30	
31	
32	
33	
34	
35	
36	
37	
38	
39	
40	
41	



INDICATED NUMBERS ONLY
EXAMPLES OF EXISTING
CIRCUIT TYPES

REVISION OF DIAGRAMS

DIAGRAM AND DRAWING NUMBER	SH.	STATUS MOD.	3.
E 27409 2001 3	1	REV.	-
— " — 3	2	"	-
— " — 1	3	"	-
— " — 1	4	"	-
		"	
		"	
		"	
		"	
		"	
		"	

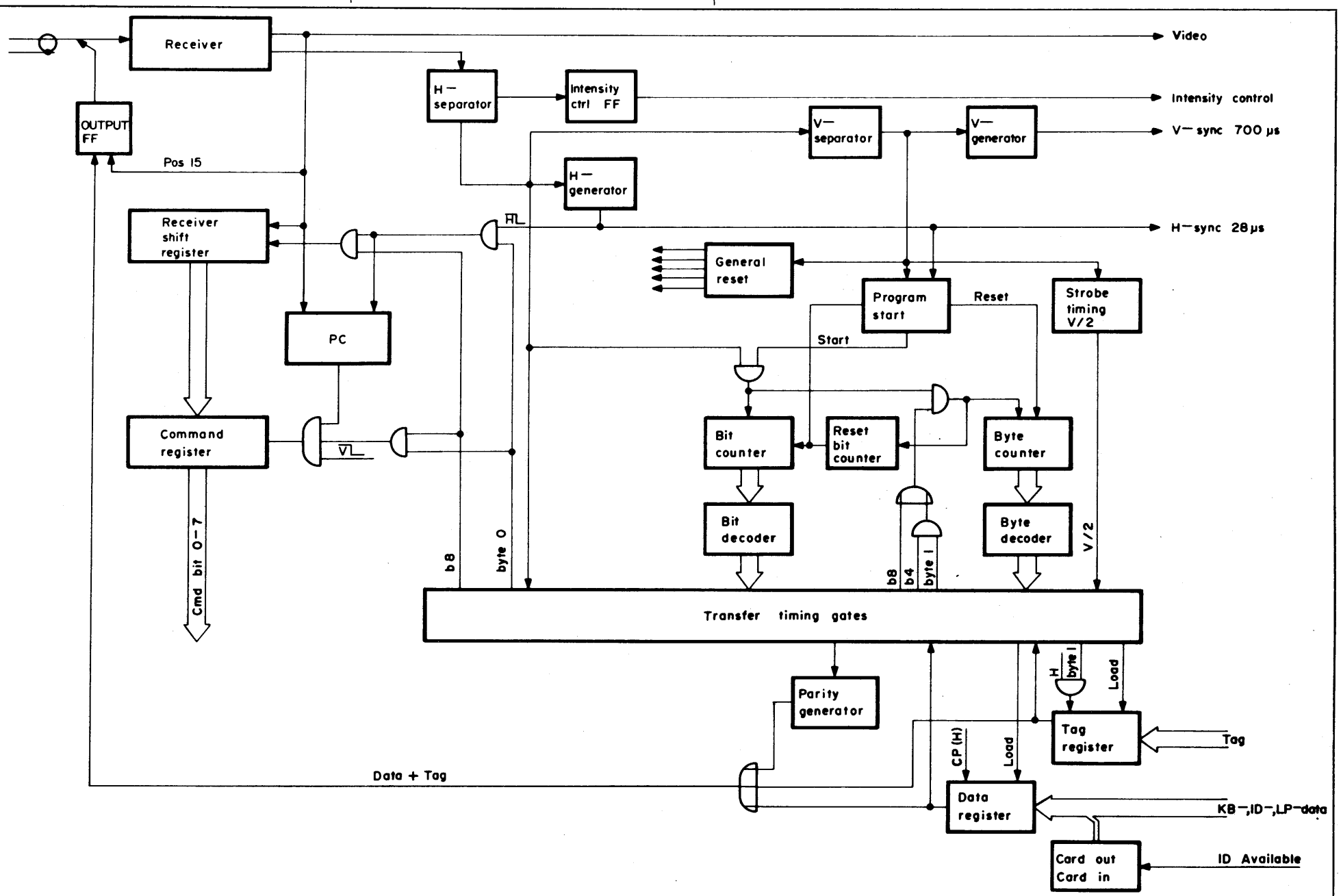
INGÅR ÄVEN I / ALSO USED ON		REG.-NR / REG. No	
FÖRST ANVÄND I / FIRST USED ON			
KONTR / CHECK 8039 AG	GODK / APPROV 0322 LB	STL SERIAL TRANSFER LOGIC	STANSAAB
SKALA / SCALE	KOPIER / COPIED 0673 L-D		E 27409 2001 3 — 4 sh — 1
UTFÖRD / ISSUED 8039 AG	740624		

This document must not be copied, reprinted or disclosed to any third party, either wholly or in part, without our consent. Any infringement will lead to legal proceedings.
STANSAAB elektronik ab SWEDEN

Denna handling får ej utan vårt medgivande betecknöras kopieras mångfaldigas eller eljest obehörigen utnyttjas. Överträdelse beivras med stöd av gällande lag.
STANSAAB Elektronik AB

OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!

This document must not be copied, reprinted or disclosed to any third party, either wholly or in part, without our consent. Any infringement will lead to legal proceedings.
STANSAAB Elektronik AB SWEDEN



INGÅR ÄVEN I / ALSO USED ON				REG.- NR / REG. No	
KONTR / CHECK	8039 Alg	GODK / APPROV	0266 LB	STL SERIAL TRANSFER LOGIC	STANSAAB
SKALA / SCALE		KOPIER / COPIED	8506 G.S.		E 27409 2001 3
UTFÖRD / ISSUED	8039 Alg	74 11 18			4 BLAD / SHEET 2

REV.1 STL

FÖRST ANVÄND I / FIRST USED ON

INGÅR ÄVEN I / ALSO USED ON

OMJAM BARRET 08 30 TERRA JB

Denna ritning får ej utan vårt medgivande bekantgöras, kopieras, mångfaldigas eller eljest användas överhuvudtaget eller meddelas till annan än den som erhållit densamma av oss.
STANSAAB

This document must not be copied, reproduced or disclosed to a third party without our consent. Any infringement will lead to legal proceedings.
STANSAAB

RECEIVER
H-SYNC SEPARATOR
V-SYNC SEPARATOR
PROGRAM START
SYNC/ INTENSITY DATA/ VIDEO
SYNC+INTENSITY
DATA + VIDEO
H-SYNC DETECTOR
V-SYNC DETEKTOR
INTENSITY CONTROL FF
GR FF
OUTPUT FF
H-SYNC GEN
V-SYNC GEN
RESET BIT COUNTER
BIT COUNTER
BYTE COUNTER
SET CMD
PARITY GEN
RECEIVER SHIFT REGISTER
PAR CHECK
COMMAND REGISTER
V-SYNC/2
CONTRAST POT
BRIGHTNESS POT
TAG+DATA+PARITY
PARITY
V - SYNC TO DISPLAY
H - SYNC TO DISPLAY
NORM/HIGH BRIGHTNESS TO DISPLAY
VIDEO TO DISPLAY
VIDEO LP
ALARM
SYSTEM READY - LAMP
I/O ERROR - LAMP
POWER ON - LAMP
SPARE - LAMP
I/M - LAMP
USM - LAMP
ID-REQ
CLICK
RETRANSM
RESET ID
RESET LP - COUNTER
BYTE 0
BYTE 1
BYTE 2
BYTE 3
BYTE 1 · BIT 0
BYTE 1 · BIT 4
BIT 0
BIT 4
BIT 8
9/sh4
10/sh4
11/sh4
12/sh4
13/sh4
14/sh4
15/sh4
16
17/sh4
18/sh4
19/sh4
20/sh4
21/sh4
22/sh4
23/sh4

INGÅR AVFN I ALSO USED ON		REG.-NR REG. No	
FÖRST ANVÄND I FIRST USED ON			
RITN. DRAWN	97-11	BOOK	
KONTR. CHECK		APPROV.	
SKALA SCALE		COPIED ON	
STANSAABS MODELL STANSAAB MODEL		COPIES	LC
UPPDRAGSNUMMER ISSUED NO		740624	

STL
SERIAL TRANSFER LOGIC
3510

STANSAAB

E 27409 2001 1 | 4 sh | 3

REV. 1

R28 - R40 = 2,2 k

READ ID
READ KB
STEP KB
READ LP
13/sh3
17/sh3
5/sh3
23/sh3
22/sh3
4/sh3
15/sh3
8/sh3
7/sh3
20/sh3
19/sh3
18/sh3
3/sh3
2/sh3
BYTE 1 - BIT 0
LP-HIT + LP-ACT
V/2
BYTE 1 - BIT 4
ID-REQ
RETRANSM
PARITY
DATA AVAIL FF
STROBE FF
LOAD
TAG SHIFT-REGISTER
DATA SHIFT-REGISTER
LP-HIT MEMORY
REPEAT + STROBE MEMORY
ID-CARD IN/OUT LOGIC
CARD OUT
CARD IN
CLICK
BYTE 2
PC/KB, ID, LP
ID AVAILABLE
STROBE/DATA AVAILABLE, KB
REPEAT, KB
LP-ACT
LP-HIT
BIT 1 / KB, ID, LP
" 2 — " —
" 3 — " —
" 4 — " —
" 5 — " —
" 6 — " —
" 7 — " —
" 8 — " —
CLICK, KB
CLICK, LP
14/sh3
10/sh3
11/sh3
1/sh3
12/sh3
6/sh3
21/sh3
9/sh3

Denna handling får ej utan vårt medgivande kopieras, mångfaldigas eller eljest obehörigen användas eller meddelas till utomstående.
STANSAAB

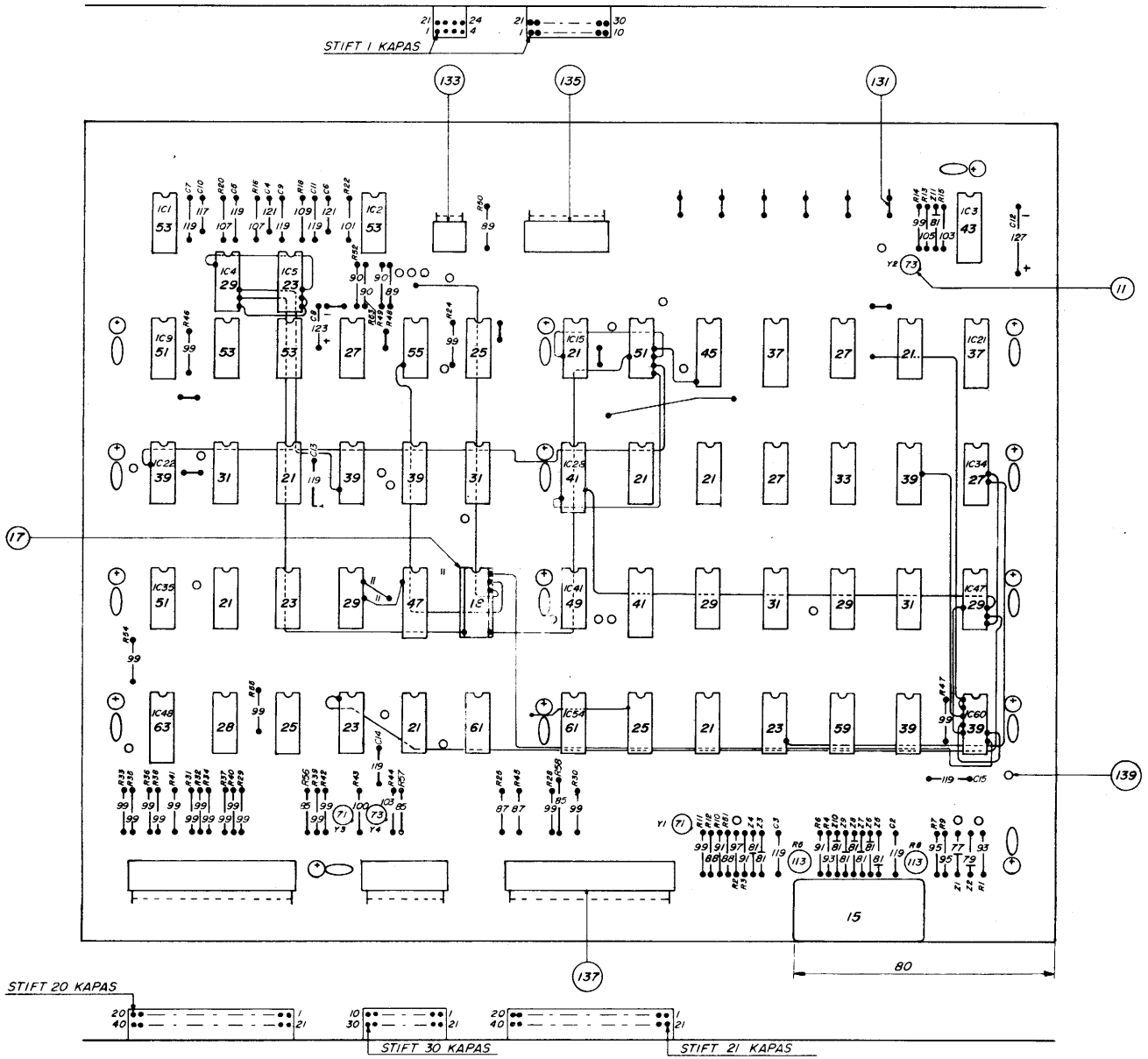
OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!

This document must not be copied or otherwise disclosed to any third party, either wholly or in part, without our consent. Any infringement will be prosecuted by STANSAAB
STANSAAB äger upphovsrätten till denna ritning
SCANDBACO VELVETEX

INGÅR ÄVEN I ALSO USED ON			REG.-NR REG. No
KONTR CHECK	1803 Alg	GODK APPROV	2366
SKALA SCALE		KOPIER COPIED	0813 LO
UTFÄRD ISSUED	8039 ATg 7406 24		

STL
SERIAL TRANSFER LOGIC
3510

STANSAAB
E 27409 2001 1
4 BLAD SHEET 4



KODNING AV KONTAKTER

P1	STIFT	20	KAPAS
P2	"	30	"
P3	"	21	"
P4	"	1	"
P5	"	1	"

FÖRBINDNING BRYTES

II = KOMPSIDA 3 ST

V = LÖDSIDA SE BLAD 2

SAMTLIGA TRÅDBYGLAR A34075 0003

MATERIAL ENL POS 140

TILLÅTEN LÖDHÖJD 3MM

STL		STANSAAB	
KRETSKORT		E27409 0001 2	2 BLAD 1
PRINTED BOARD ASSEMBLY			

8059	0246	8039	740426
ALg	L.B.	ALg	
1:1	8477		
	805		

E27409 0001 2

REV 1 740813 | 2 741022 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |

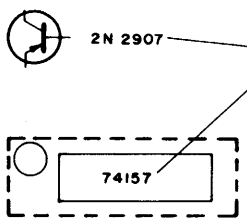
STL

STANSAAB Elektronik AB SWEDEN

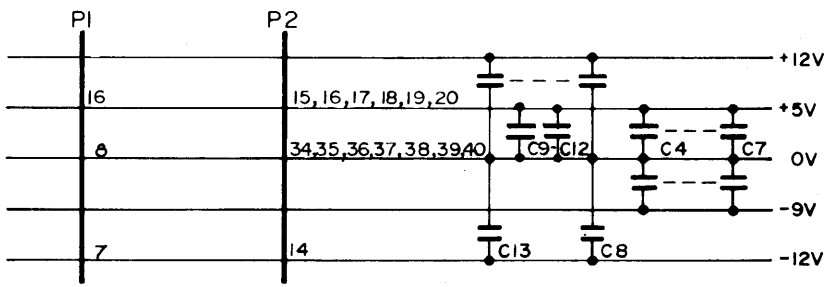
IF IN DOUBT: ASK!

Denna handling får ej utan vart medgivande bekantgoras, kopieras, mångfaldigas eller eljest obehörigen utnyttjas. Överträdelse bevakas med stöd av gällande lag
STANSAAB elektronik ab

TEST POINTS	
NR	SIGNAL
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	
32	
33	
34	
35	
36	
37	
38	
39	
40	
41	



INDICATED NUMBERS ONLY
EXAMPLES OF EXISTING
CIRCUIT TYPES



REVISION OF DIAGRAMS AND ASSEMBLY DRAWING

DIAGRAM AND DRAWING NUMBER	SH.	STATUS MOD.	–
E27411 2000 3	1	REV.	1
—"— 2	2	"	1
—"— 7	3	"	1
		"	
		"	
		"	
		"	
		"	
		"	
E27411 0000 3		"	

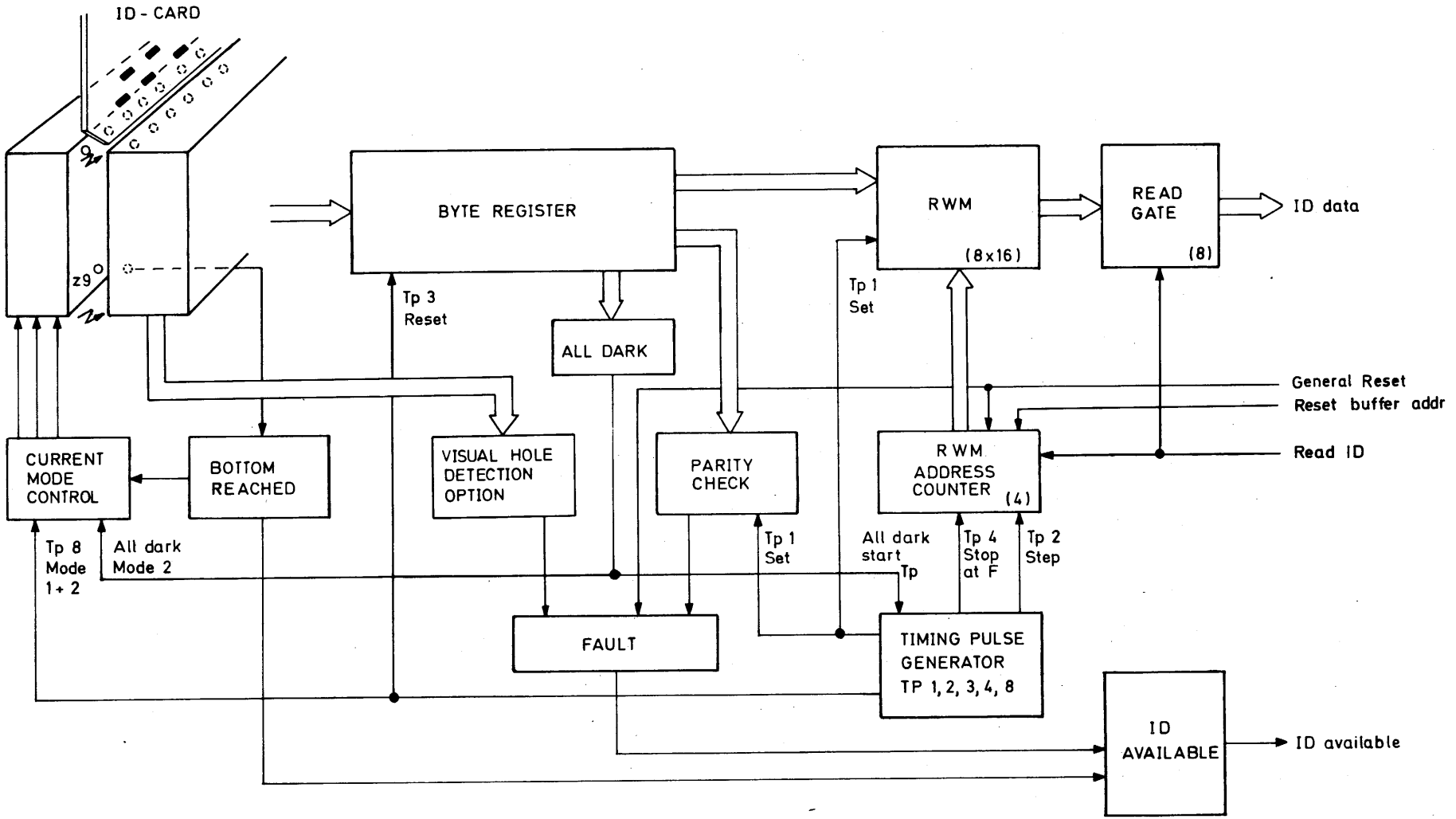
INGÅR AVEN I ALSO USED ON		REG. NR REG. No	
FÖRST ANVAND FIRST USED ON			
KONTR CHECK 9748	GODK APPRO. 8266 LB	ID Identification Device	STANSAAB
SKALA SCALE	KOPIER COPIED 9328 GM		E27411 2000 3 — 3 sh 1
UTFÖRD ISSUED 9748 MP	730628		

This document must not be copied, reprinted or disclosed to others without written permission from STANSAAB elektronik ab. Any infringement will lead to legal proceedings.
STANSAAB elektronik ab SWEDEN

Denna handling får ej utan, vårt medgivande bekantgöras kopieras mångfaldigas eller eljest obehörigen utnyttjas. Överträdelse beivras med stöd av gällande lag.
STANSAAB Elektronik AB

OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!

This document must not be copied, reprinted or disclosed to any third party, either wholly or in part, without our consent. Any infringement will lead to legal proceedings.
STANSAAB Elektronik AB SWEDEN



REG. NR / REG. NO				ID Identificaton Device	STANSAAB
KONTR / CHECK	9748	GODK / APPROV	0226 LB		E27411 2000 3 — 3 sheet 2
SKALA / SCALE		KOPIEN / COPIED	9328 GM		
UTFÖRD / ISSUED	9748 MP		730628		

REV.1 | 74113

FÖRST ANVÄND I / FIRST USED ON

INGÅR ÄVEN I / ALSO USED ON

OMJAM BARRET 33 JOT 18

Light censors

+5V +5V

V11

R11 39k

R13 6,8K

R12 39k

-12V

V12

2B50C

-12V

Current generators Lighternitters

2N2905

Y1

R101 120Ω

Z100

R120 2,2k

33nF C3

R102 120Ω

Z1 Z2 Z3

Y2

R103 180Ω

2N2905

Z101

R121 2,2k

Z4 33nF C2

R104 120Ω

Z5

R105 120Ω

Z02

Z6

Y3

R106 100 Ω

2N2905

Z7

R122 470Ω

Z8

Bottom reached cens.

R107 180Ω

Z9

Z1-8 = TIL 31

0V

+12V

+5V

22 24 8 12 16 21

Byte register

Parity Generator

Current Mode Control

All Light

All dark

20 Parity 74180 Even in Odd in 3 even 3 odd

0V

+V2

Card detected

+V1

Fault

TP Generator

R115 22k

C1 1nF

R9 680 Ω

R109 1,2k

R108 1,2k

Bottom reached

ID Available

Z1 220Ω R10

R118 1,2k

Read gate

Interface

RWM Adr. counter

Cout 74193

RWM

MC RWM WE 16x4 Bit 7489

R1-8 1,2k

R119 12 k

R116 2,2k

P2

Reset ID-Buffer Add.

General Reset

B1

B2

B3

B4

B5

B6

B7

PC=B7

Read ID

ID-Available

OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!

This document must not be copied, reported or disclosed to any third party, and used for any purpose without our consent.
STANSAAB Elektronik AB SWEDEN

Denna handling får ej utan vårt medgivande kopieras, återges eller på annat sätt obehörigen användas. Överträdelse medför skadeståndsskyldighet.
STANSAAB Elektronik AB

REV. | 7406 28

SCANDIAZO VELVETEX

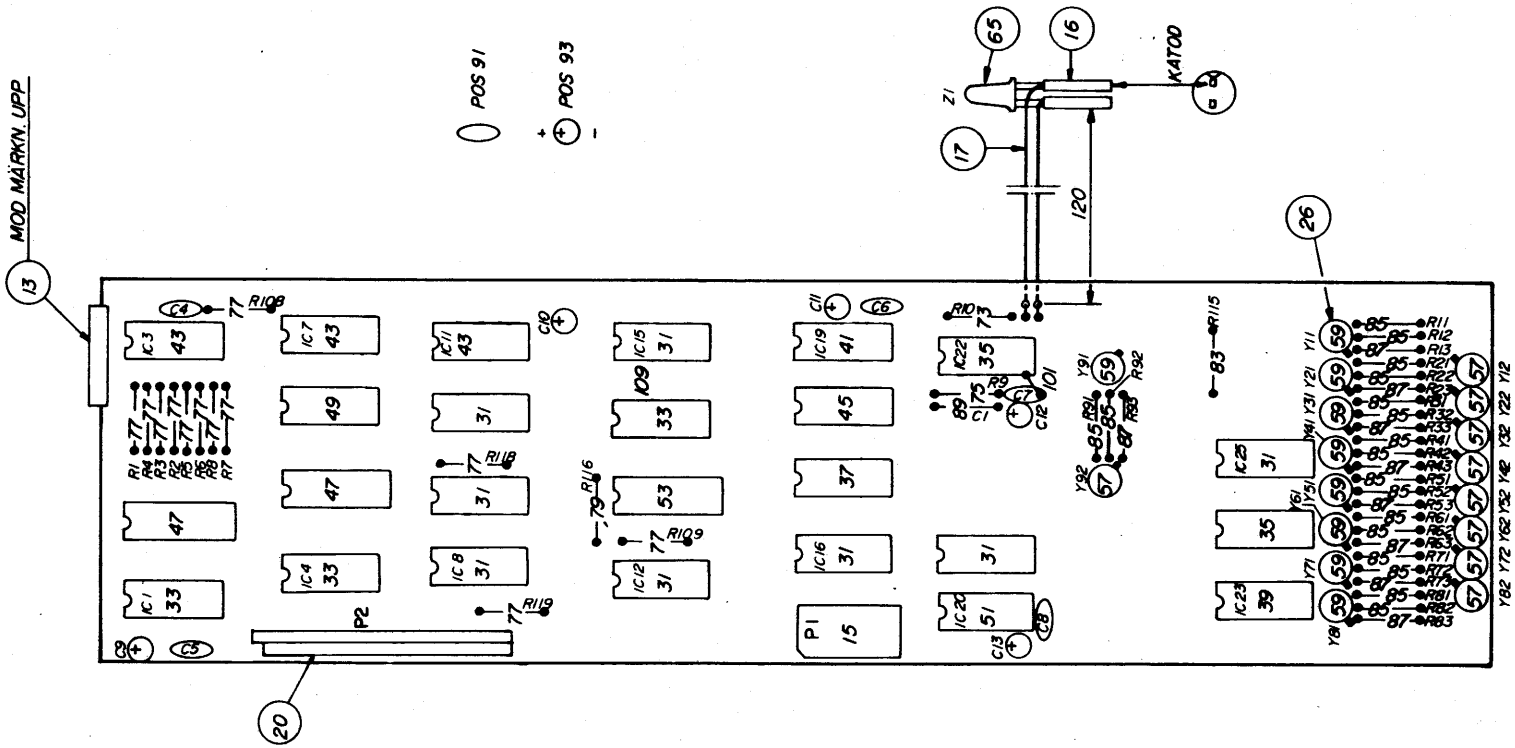
KONTR CHECK	GODK APPROV	0266 28	ID IDENTIFICATION DIVICE
SKALA SCALE	KOPIER COPIED	8839 BAO	STANSAAB
UTFÖRD ISSUED	9748 MP	7306 28	E 27411 2000 7 — BLAD SHEET 3

Denna handling får ej utan vårt medgivande bekantgöras kopieras mångfaldigas eller eljest obehörigen utnyttjas. Överträdelse beivras med stöd av gällande lag.
STANSAAB Elektronik AB

OM NÅGOT ÄR OKLART: FRÅGA!
IF IN DOUBT: ASK!

This document must not be copied, reprinted or disclosed to any third party, either wholly or in part, without our consent. Any infringement will lead to legal proceedings.
STANSAAB Elektronik AB SWEDEN

REV.												
1	740628	740913	740927	741204								E27417 0000 3



POS 57 MONTERAS I FIXTUR
HÖJD ÖVER KORTET 7,5 MM.
STIFT 40 I POS. 20 MONTERAS EJ.

TILLÅTEN LÖDHÖJD 3MM

REG.-NR / REG. NO			
KONTR / CHECK		GODK / APPROV	
SKALA / SCALE	1:1	KOPIER / COPIED	9569 LeH
UTFÖRD / ISSUED	9748 MP		740305

LOGIKKORT
LOGIC ELEMENT BOARD
KRETSKORT
PRINTED BOARD ASSEMBLY

STANSAAB

E27417 0000 3

FÖRST ANVÄND I / FIRST USED ON

INGÅR ÄVEN I / ALSO USED ON

OMJAM TERRAB MAJMO

P2 P1

Read KB 16
Ret 36

Click 18
Ret 38

Read ID 3 13
Ret 23 33
ID-available 4 12
Ret 24 32

B1 13 6 1
Repeat 26 R
B2 12 7 2
B3 11 27 3
B4 10 8 4
B5 9 28 5
Ret 28 9
B6 8 29 6
B7 7 10 7
B8 30 8
B9 (PC) 6 11 9
Ret 26 31
Strobe/Data available 15 S
Ret 35

Reserv 1 5
Reserv 2 25

0V 21
0V 22
0V 27
0V 29
0V 30
0V 31
0V 32
0V 33

+5V 16
0V 36
+5V 17
0V 37
+5V 18
Ret 38 21
+5V 19 2
Ret 39 22
+5V 20 3
Ret 40 23

USM ⊗ 20
USM ⊗ 40
IM ⊗ 19
IM ⊗ 39

Click

Oscillator

Aut repetition

Bounce blocking

Row counter

Column counter

Output register

Parity generator

Interface

Shift mode FF

1 of 16 decoder

Selector

Keyboard switch matrix

C1 C2 C3 C2,3 C3,2 C6,4 C7,5 C9,8 C3,8 C3,5 C A,B C C,D C E,F

RO RO,F SHIFT KEYS

5V 75 mA

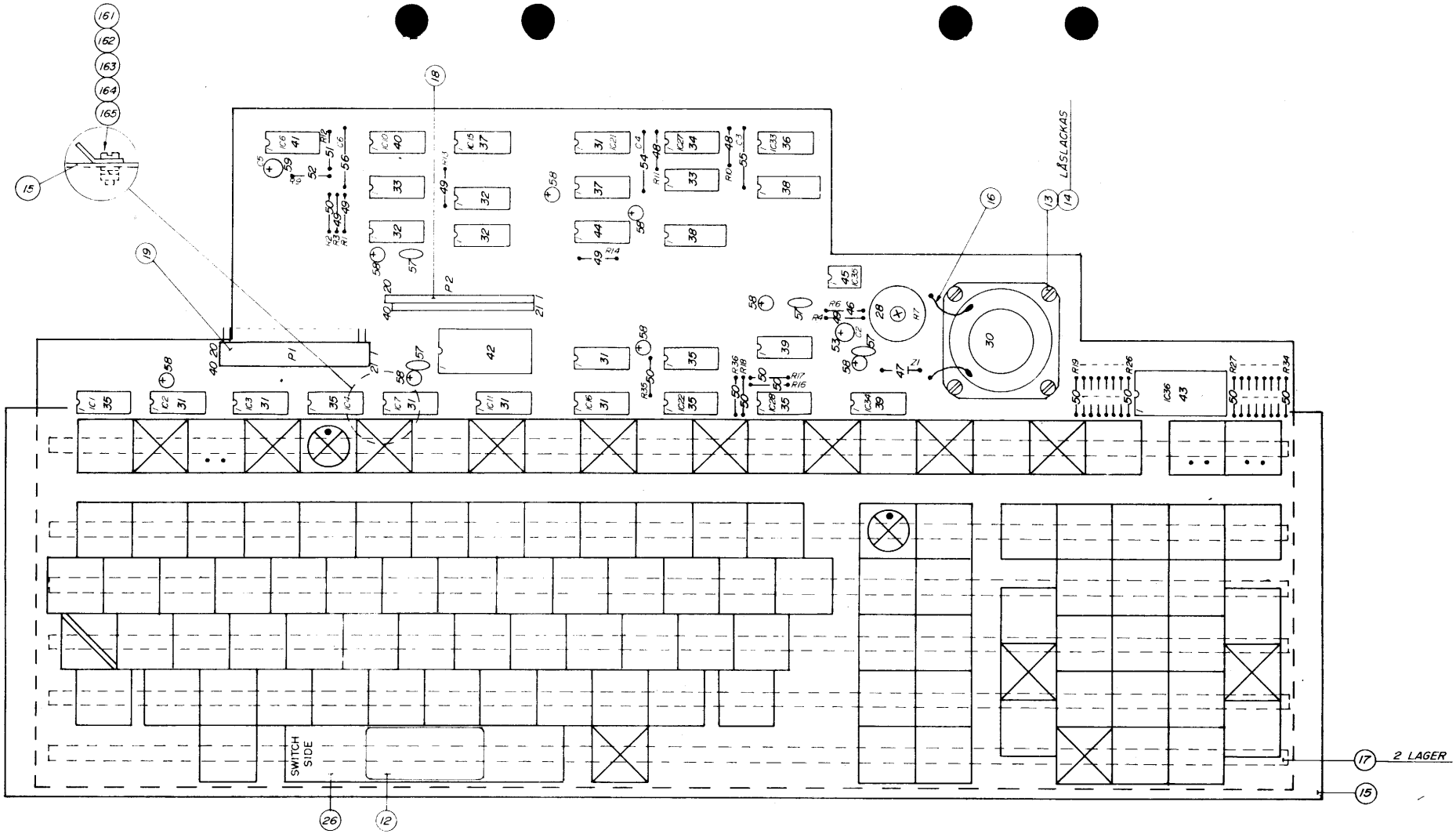
5V 75 mA

This document must not be copied, reproduced or disclosed to any third party, wholly or in part, without our consent. Any infringement will be prosecuted. STANSAAB sidtrade ab SWEDEN

OM NÅGOT ÄR OKLART, FRÅGA!
IF IN DOUBT: ASK!

Denna handling får ej utan vårt medgivande, bekräftelse, kopieras, mångfaldigas eller utan obehörigen utnyttjas. Överträdelse beivras enligt "STANSAAB" riktlinjer.

INGÅR ÄVEN I ALSO USED ON				REG.-NR REG. No.	
KONTR CHECK		GODK APPROV	02CC 2.28	STANSAAB	
SKALA SCALE		KOPIER COPIED	8839 BA6	Key board	
UTFÄRD ISSUED	9748 MP	74 0115		E00101 2000 1	BLAD SHEET



= POS 27

= POS 21

= POS 25
FÄRGPUNKT MELLAN STIFTEN

= POS 23

KODNING AV KONTAKTER :
P1 STIFT 21 KAPAS
P2 STIFT 40 - " -

SKUMPLAST POS 160 MONTERAS PÅ INSIDAN AV POS 149 :



MONTERING AV TANGENTKNAPPAR : SE LAYOUTRITNING '
MOUNTING OF KEYTOPS : SEE LAYOUT SPEC.

REG NR / REG NO			
KONTR / CHECK	9746	GODK / APPROV	0126 CB
SKALA / SCALE	1:1	KOPIER / COPIED	9696 HP
UTFÖRD / ISSUED	9748 MP	740502	

TANGENTUR MONTERAD
KEYBOARD MOUNTED

STANSAAB

E00101 0000 2

REV. 1 740530 KB

STANSAAB ELEKTRONIK AB SWEDEN

E00101 0000 2

STANSAAB Elektronik AB

FÖRSTÅS ANLAND
FIRST USED ON

INGÅR AVEN I
ALSO USED ON

OMJAM PARRET ÄR 10E 18