P800 Hardware Guide

Table of contents:

1. Introduction
2. Belier family
5. The P851
7. P800 Hardware Documentation
8. The GP bus
9. Micro programs
12. Interrupt
14. The MCU's
15. V24
18. Line printers
20. The X1215 disc
21. The Floppy disc
23. The I.O. Processor
25. The Trap
26. The MMU
27. The Instruction Set
28. The Cassette
31. IO instructions for cassette, tape, disc & floppy
34. Tape and disc modulation systems
36. Testing the PERTEC tape drive
38. Programming examples
45. Data Communication
47. The I.E.C. Bus
55. Integrated circuits
67. New in 1980

For Training use only! Philips-S&I-IDS-Service/Training
770830, 780829, 790117, 790420, 800325.

P800 Hardware Guide.

Introduction.

The purpose of this book is to help you to find your way in both the P800 hardware and in the P800 documentation. It also covers some aspects of the software that may be of interest to the service engineer.
Wherever a CPU is mentioned we talk about the P856. Reference to any other machine is explicitely stated. Most of the information is valid for P852-P856-P857, some of it also for the others.

The P800 Family.

The 1st 16-bit machine produced by Philips Data Systems was the P880. It was never used by S&I, it is completely different from the newer 800's Somewhere around 1972 the Sagittaire family was introduced. It's members were P850, P855 and P860. A technical change upgraded the P855 to the P860 level, the name P860 is not used anymore. The P850 was a rather small machine, it is not produced anymore. Around 1975 the Belier family started its life. Hardware wise it is completely different from the Sagittaire, the software is 95% compatible. The P852, P856 and P857 belong to the Belier family. The P851, available in 1977, is again fully different but still fully compatible with the other Beliers if some precautions are taken Software compatibility of the P851 depends on the type of memory that is used, either RAM or ROM.

French:	English:	Dutch:	German:
Sagittaire	Sagittarius	Boogschutter	Schütze
Belier	Aries	Ram	Widder

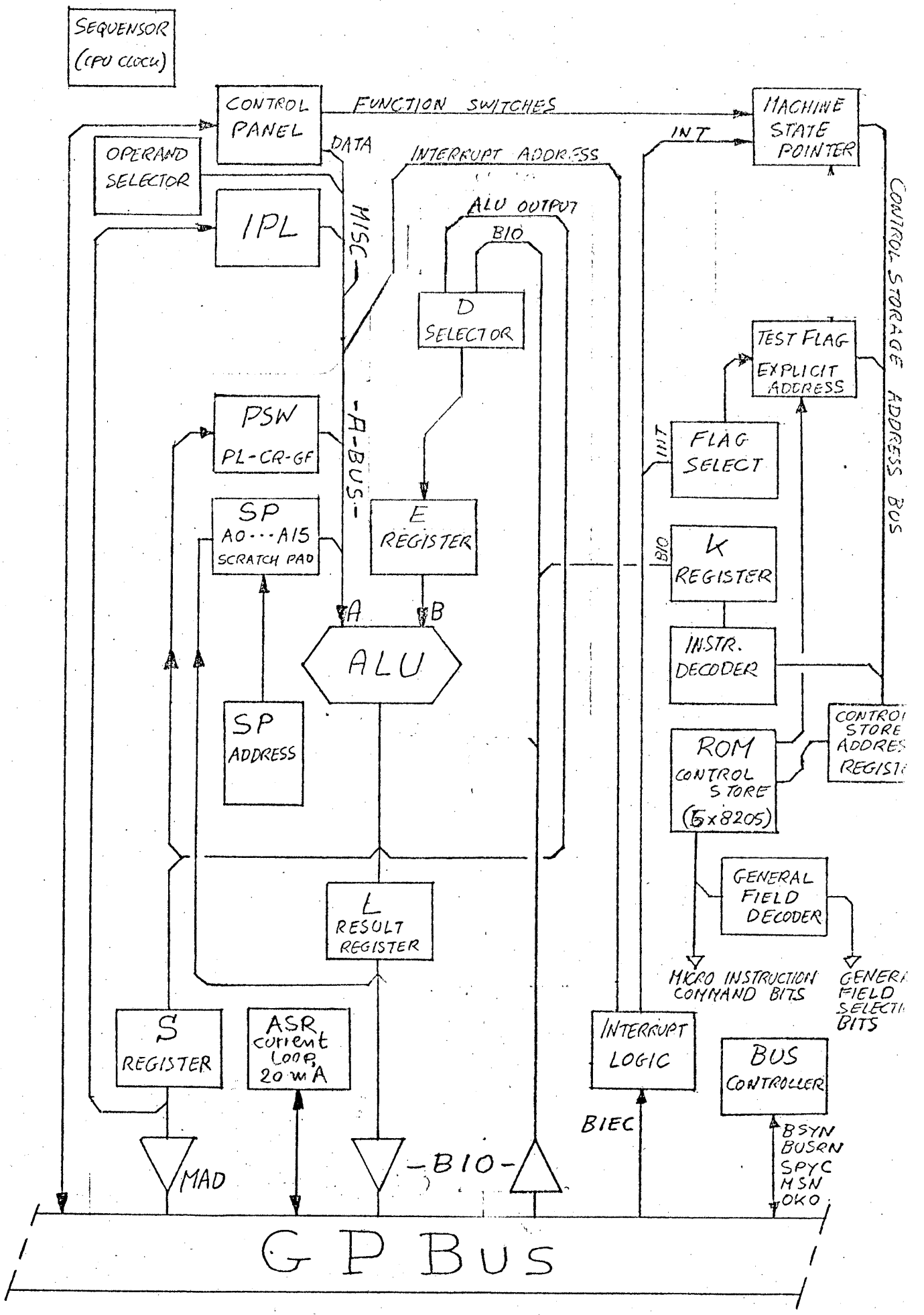
The Peripheral Equipment

Most of the input-output devices can be used on all P800 models. It only depends on the fact if a control unit for a certain device is available for a certain family of machines. If not there is a possibility to use a so-called Bus Translator to connect a control unit to a CPU of another family. This book does not pretend to give full information on all peripheral devices; you will always need both the device manual and the control-unit manual if there is any problem with a device.

also see: page 67!

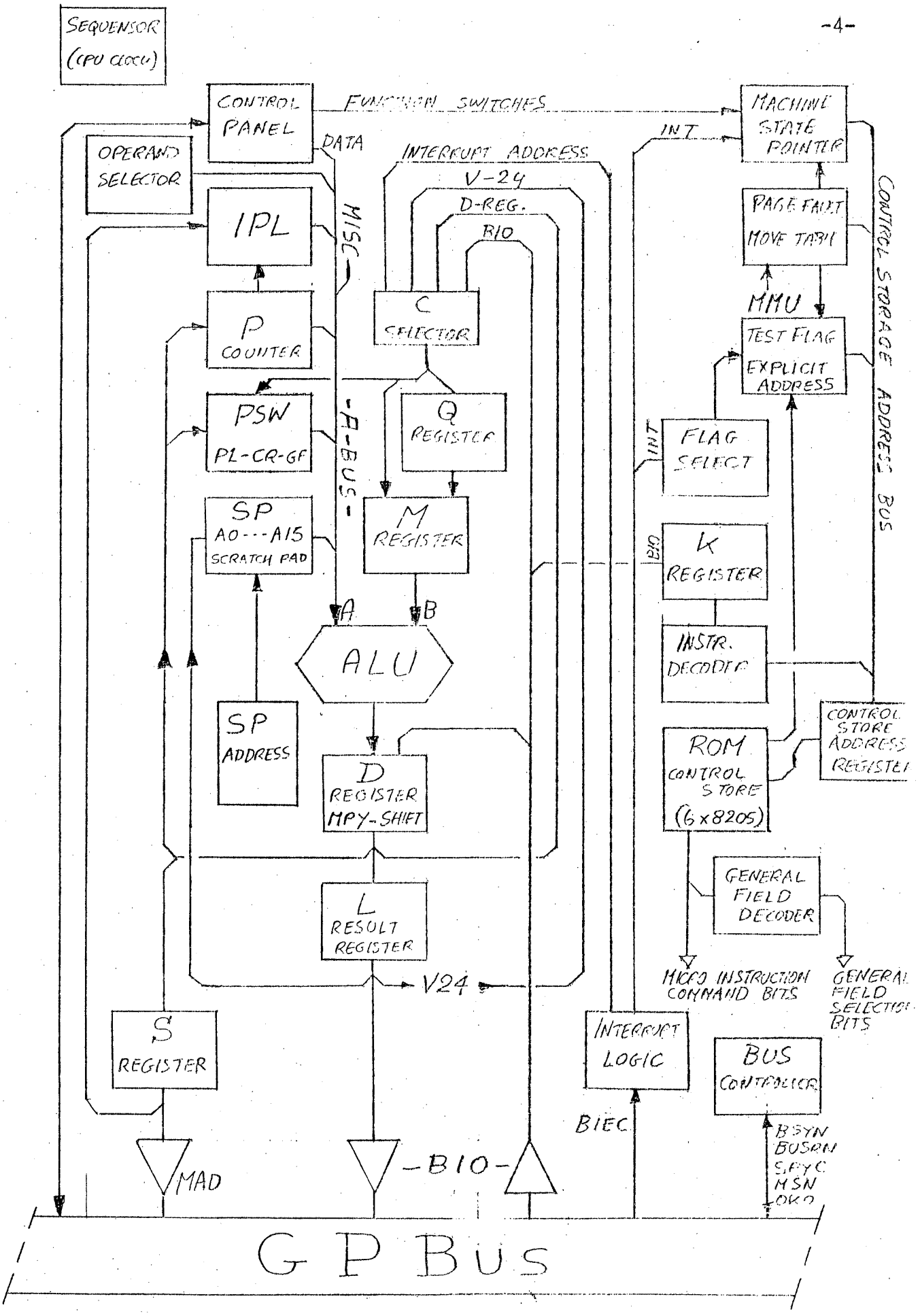
The Belier Family.

The P852, P856 and P857 all use the same General Purpose Bus to interconnec the main parts of the system. They use only one 19 inch printed-circuit board for the CPU, which is micro-programmed. They all have 63 hardware interrupt levels.
The P852 has a restricted instruction set. There are no instructions that handle operands longer than one word.(No ML, MS or double shift)
Also, multiply and divide can only be done by software(Trap Action)
The P856-857 has an instruction set which includes multiply & divide. (Multiply & divide is an option for the P855). Some instructions that were needed for the P855 interrupt system are no longer used.
The difference between the P856 and P857 lies only in the options availabl for the CPU. P852 and P856 use the same small operators panel. The larger panel of the P857 is available as an option on the P856. This panel makes stopping on a preset address possible. It also contains some test-programs in ROM. These test the most important functions of the CPU and also some memory and a little bit of input-output. There are no other options for th P856 but on the P857 the MMU and the FPP is possible. (MMU=Memory Managemer Unit, makes it possible to connect 128K. FPP=Floating Point Processor, thi is hardware to handle floating point numbers as used in Fortran programs)
The memories for the P856-P857 are available in different sizes and speeds. Because of the micro-programming system that is used the influence of the memory speed on the machine speed is not as great as you would think.
If two 16K modules are used as one 32K module they can be set up to work in interleaving mode. In all P800 memories bit 15 of the address is used t select the left- or right half of a word. With the interleaving technique bit 14 is used to select one of two modules. If now a consecutive number of memory words is accessed they will be found in either one of the modules alternatingly. This makes it possible to start a cycle on the next word when the machine is still thinking about the previous one. Of course this is only effective if the words are in sequence; this is the case in about 50% of any program.
If you compare the block diagrams on the next two pages you will see that the P852 has no Q register; in this register the double-word operations are carried out. You will see more differences, the larger CPU could only be built on the same board by using more complex LSI(=Large Scale Integrati and a multi-layer printed circuit board (6 layers).
One small difference is that the P852 has a "current-loop" interface for a typewriter, and the P856-P857 a "V24" interface. The latter interface makes it possible to use the "Break" key as a "control panel interrupt"



P852 GENERAL BLOCK DIAGRAM

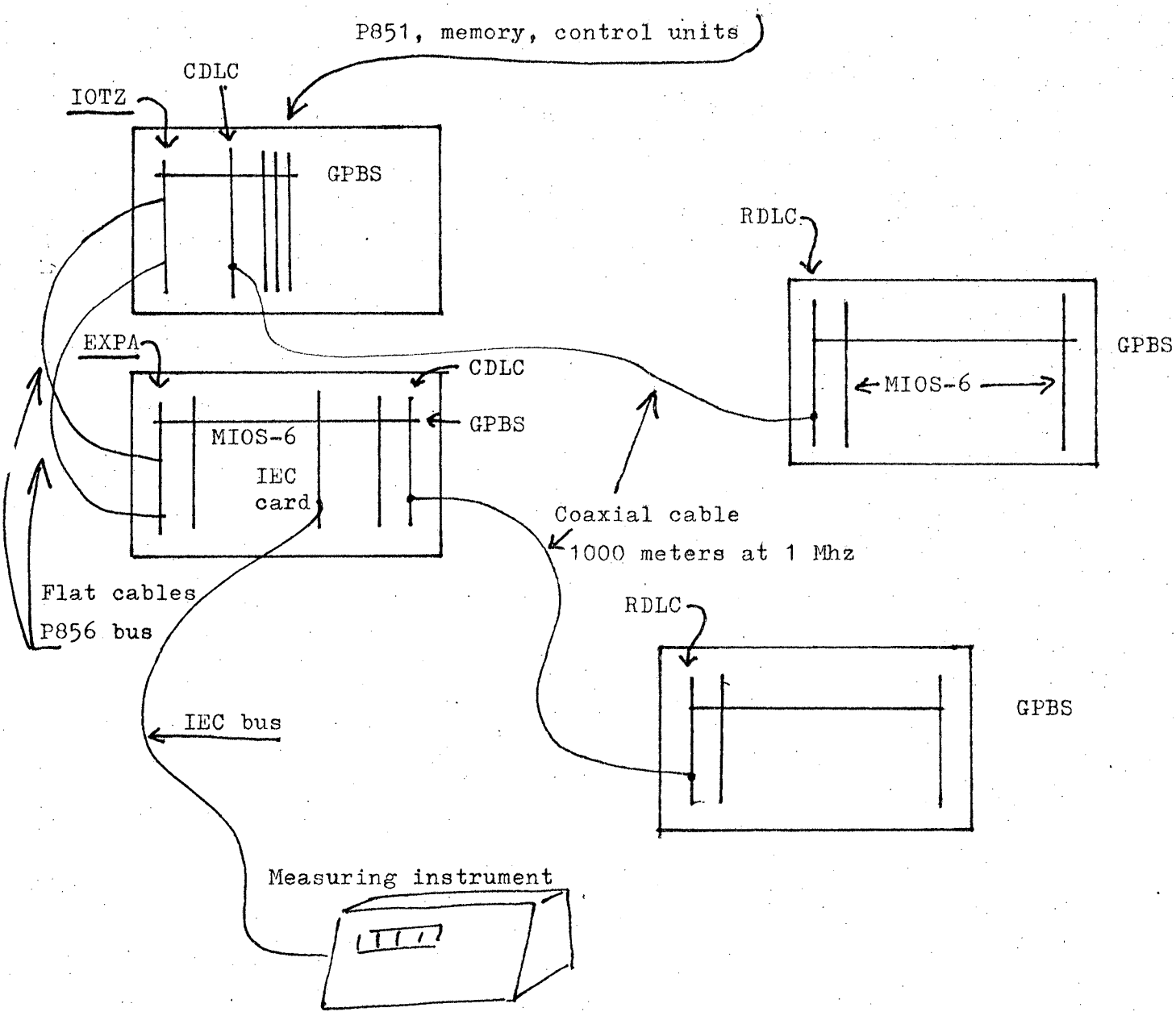
-760521-



P856-P857 GENERAL BLOCK DIAGRAM

The P851

Altho the P851 has exactly the same instruction set as the P856 it is a completely different machine. The CPU is built on a "double EURO board", this board is mounted vertically in a "EURO rack". Vertical mounting gives a more effective natural cooling. This is very often sufficient because the P851 uses very little power. The EURO racks are made by many different manufacturers, they are already used frequently by future P851 customers. These people can implement the P851 very easily in their system. All memories, control units, etc. are also on the same EURO boards. They are interconnected by a bus that is not exactly the same as the Belier GP bus, but slightly simplified. It is called the GPBS. In some documents the Z-bus is mentioned, this means the same GPBS.
The small size of the P851 is possible because specially made integrated circuits are used (Elcoma-Nijmegen). The LOCMOS technology makes both very low power consumption and high noise immunity possible.
The P851 can use both RAM and (P)ROM memories (Random Access Mmemory, in th text read-write memory, (Programmable)Read Only Memory).
The P851 can be used in the micro-computer applications field. Once a program is put in a PROM it will stay there forever. Of course the program needs working areas in RAM. A P851 works as a P856 (only somewhat slower) when all its memory is RAM. The difference in speed is a factor 2 or 3. The RAM will lose its contents if no battery-back-up is used when there is a power drop. Of course there is no such problem with the PROM.
A special type of Modular Input-Output System is made for this GPBS, the MIOS-6. It uses the same double EURO boards as the other P851 cicuitry. Each MIOS_6 board is addressed directly by a WER or a RER instruction, without the use of a MIOS controller. The 2-step system as used in the older type of MIOS is no longer necessary. Only in the Analog part this is somewhat more complex. More about this in the chapter about MIOS.
By using EXPA (Bus Expander) more EURO racks can be connected. Also the standard P856 GPB can be connected, or the P856 can use MIOS-6 in this way.
The main difference between the buses is the interrupt structure: the Belier uses a 6-bit interrupt bus, the P851 a 20 interrupt line bus. For this reason the Belier can recognize 2^6 interrupts, the P851 only 16.
Finally the P851 has built-in test programs, similar to the P857 operators panel.



(as used in PM 4410)

RDLC: Remote
CDLC: Central
data link controller

The P800 Hardware Documentation.

The CPU service manuals give a sufficient description of the CPU only. As they say nothing about the CPU as it is used in a system they are not very useful for service purposes. The policy of system repair is nowadays change the faulty board and send it to the factory for repair. In most cases the error on a very complex board is hard to find; you need special test equipment and even then repair may be a time consuming job. On other boards, as the memory, specially selected I.C.'s are used which are not normally available. So repairing a system is no longer finding a bad component on a board, but finding a bad board in a system. What you need is a description of the system. This is difficult because every system is different. The book that comes closest is the P800 Interface and Installati Manual (5122 991 26942). It tells you how to set up the jumper wires on most boards, how to connect the cables, etc. It does not give many details about the boards themselves or about the programming, or the operating of a system.

If you have to trouble-shoot the input-output devices you will need the manual of the device itself and the control-unit manual. In some cases you will also need the description of the I.O.Processor and/or some special tools. The control-units for PR (Papertape Reader), PP (Papertape Punch) and V24 are described in the MCU3 manual. The V24 interface that is part of the P856-P857 CPU is slightly different from the one on the MCU3 board. It's description is in the CPU manual.

The LP (Line Printer) and CR (Card Reader) are to be found in the MCU2 manual. The other CU's do not share boards, they may be found under their own name.

90% of all errors are relatively easy to find because they can be tracked down to some specific part of the system. Sometimes you have a problem where all test-programs run perfectly and still the customers program does not go as it used to. Then you need a good description of the system, and you must be very sure that nothing was changed in the customers program It appears to be very difficult to keep the system description up-to-date! And remember that it is possible that a program with some error in it has been running perfectly for a long time. The error then only shows when a very specific set of conditions is there. You may need help from the system programmer to find such an error.

The GP Bus.

The GP (General Purpose) Bus is the highway for data and control in the Belier machines. The Sagittaire uses the somewhat different I.O. Bus: The GPBS is already mentioned on page 5.

Going from one bus to another is possible through the use of translator boards.

The main advantage of the GP Bus is that it uses a conversational system If the CPU wants to use the memory, it tells the memory so. When the memory is ready, it answers back to the CPU. At the same time a Bus Controller takes care of reserving time on the bus to enable the data transport.

In a similar way communication between CPU and peripherals, or memory through IOP to peripherals is organized. Synchronization is more or less automatic in this way: memory modules with different cycle-time may be used in one system.

The response time of a device depends on the length of the GP Bus: If the control unit board is in the same cabinet as the CPU the GP Bus is very short. Depending on the cabinet model it has a maximum length of about 45 c This gives a delay that may be ignored. When Bus Cables are used this is different. The maximum length of the bus is 14.5 metres, the delay is about 7 nS/metre, and the delay works both ways. Added to this is the delay caused by the CU circuitry. So there is no fixed timing for the response. For this reason the CPU (Or rather the Bus Controller) has a time-out cicuit. If the device does not respond within 6.5 uS the device address is said to be unknown. The condition register of the CPU is set to 3.

The data travels on a 16 wire two-way bus, the BIO. The memory and the I.O. devices share one address bus, the MAD bus. Also the I.O. function bits go on the MAD bus. There are 18 MAD lines, to enable the P857 to address 128K memories.

The bus cables go from one extension shelf to another. In the last one the terminator board should not be forgotten. It is not needed in a system without bus cables; it may cause marginal operation if left out in larger systems.

The wiring from one equipment shelf to another is very simple; see installation manual. There is one exception: the cassette drive.

One shelf houses both the cassette control unit and the drives. The bus cables which are normally on the left-hand side of the board go on the back of this shelf. There are may ways to put the connectors on in the wrong way. This may burn out the cassette controller, the CPU board and the cables! So, be careful!

Micro-Programs.

The P856 Service Manual (5122 991 26953) is not very easy to read. To help you find your way we will analyse the execution of:

ADK A5,25

All page nrs. etc. refer to the above mentioned manual.
First go to figure 1-8. If the machine runs normally without interrupt you will go through the diagram starting at the upper left corner. RUNF=1, C.PANEL=0, IR=0. KRY means K-Ready, in other words, the K-Register does contain an instruction. Initially this is not so, KRY=0, so you get to the block FETCH at the bottom of the page. The digits 1-13 refer to the next flow-diagram. Note the number 016,
this is the address of the Fetch Micro-Instruction.
This Micro-Instruction can be found 4 times in the book: in the tables 2-4A, 2-4B, 2-5A, 2-5B. A & B refer to P856 & P857. Table 4 or 5 is either in Mnemonic form or in bits. The only difference I found is on 0E1 and 0F1, the micro-instructions for MVF & MVB!
In the first column of page 2-23 we look up line 016:

	SNA	NA	A	ADL	C	M	Q	S	P	CT	B	SEQ	R	GP	CR
016	11	111111111	00000	00000	11	10	11	10	11	0	000	10	0	11100	00000

The letters are explained in table 2-2, page 2-8.
SNA 11 means "Machine State Mode", see page 2-9.So the next address depends on the machine state, which is what figure 1-8 shows.
Nothing goes to the A-bus(see little table on page 2-65)
The two "C" bits can be found on page 2-56. This is the complete ROM for the P856-P857. Look for part N13, signals muC0 and muC1. Their destination is JJ, meaning figure 2-8JJ, or page 2-63. We are now interested in the upper part of the page, the C-selector. Both muC0 and C1 are high signals what this means can be found in a table on page 2-63. The signal GCSELN is high because it is not generated. (See table 2-3, page 2-13, muGP= 11100, so only GFETCHN is generated) You can see on page 2-63 that the components K7, J7, K8 and J8 select BIO 08---15.
On page 1-23, figure 1-13 you find micro-instruction 16. If you read righ as right you may understand why BIO 08---15 is selected.
Not everything that happened yet can be found here. For instance, starting a memory cycle. This is done with the signal TMRN, the memory must send back TRMN as described on page 2-3. To find these signals in the logic drawings you must refer to table 2-14 on page 2-53. You find there that both signals can be found on drawing TT, which is on page 2-71.
The TMRN signal is in the lower right hand corner. It depends, amongst other signals on muTMRN which comes from the ROM, page 2-56.

Here you find it as an output from component Q13, one of the few that is used in a negated form. In other words, true if zero. Now go to page 2-8, table 2-2, and look up muTMRN. It is bit 32 from the ROM. On our previous sheet we have put down the bit pattern for word 016; bit 32 is the 1st bit of the B field, which is 0. So, muTMRN is true.
Many of the signals that come very often are taken for granted in the flowsheets as the one on page 1-23, but if you really want to, you can find if and when it is generated.
Have another look on page 1-23. The line:

FRUNZO if RUN on pr. coin

means: The Run FF is reset if the program was running and there was a "preset coincidence". With the extended operators panel you can set up an address in the upper row of switches. If the rightmost switch is in the "ACC" position the program will stop if the address during the FETCH cycle is the same (coincident) as the one in the switches.
Also shown is the incrementing of P and S. Try to find how this is done in the hardware.
We found GFETCH already, CTZ16 means "16 to CounTer", check this.
Now go back to figure 1-8, "Machine State".
The only thing that changed (if there was no pr. coin) is KRY.
The KRY flip-flop is on the left edge of page 2-57. GFETCH is on the reset input, but because the ouputs are used in an upside-down manner, it acts as a set signal. If KRY=1 on page 1-19 we go to PLA. These letters stand for "Programmable Logic Array". "Programmable" means that the relations between in- and outputs can be determined by the user. The user in this case is the CPU manufacturer: it is not possible for you to program this logic array. The component can be found at the bottom of page 2-55.
How the inputs control the outputs is tabulated on page 2-26/27.
Our instruction, ADK A5,25 is now in the K-register. In bits it looks like:

```
0 1  4 5 7 8       15
0 0010 101 00011001
```

Important for the PLA are K0 and K3. You find them on line 13. The output, which is the RAM address, looks like /20. Line 20 on page 2-17 shows us that this micro-instruction has something to do with AD. All the details are on page 2-23:

```
   SNA     NA       A     ADL   C  M  Q  S  P  CT B SEQ R  GP    CR
020  11 111111111 01100 00110 11 10 11 10 11 0 000 10 0 11100 10100
```

SNA and NA happen to be the same as with FETCH. This is so because we happened to pick an instruction that uses only one ROM word for its control.
The "A" field controls the A-Bus. Have a short look on page 2-2, then goto figure 2-8LL/MM (page 2-65/66). At the top of MM muA2, 3 and 4 come in.

Their state is 100, this goes to the scratchpad address selector. How this one works is shown in the table in the middle of the page;in our case the address from the R3 field is used to address the scratchpad.
The ADL field controls the ALU. The ALU is on page 2-61, the control on 2-6
Our ADL field is 00110, try to find what this does on page 2-60.
Not so easy? Then go to page 2-30 and following. In the table 2-8 you find our bit combination on address 05. We get the outputs on the ALUS-N lines: 0110. Because these lines are negated we should read: 1001.
In the ALU function table (page 2-32) we see that this really tells the ALU to do A plus B, if muADL4 is 0, which it is. The A-operand comes from the A-bus, the B-operand from the M-register as shown on page 2-30.
You remember how the rigthtmost byte on the BIO went to M during the Fetch cycle? The only thing that remains to be done is, put the sum back into the scratchpad and set the condition register.
On page 2-62 you find the way from the ALU through the D-selector to the L-register. From there data goes directly to the scratchpad. Control of this lot comes from the same circuits that control the ALU itself, on 2-60.
Try to find out what page 2-67 is doing (the CR).
The easy way to find all this is to go to page 1-25 and see there what microprogram 020 is doing. Of course this only tells you what, not: how.

Note the fact that all logic drawings of the CPU are on 17 pages.
The P855, without multiply/divide takes over 100 pages. This illustrates the power of micro-programming.

Interrupt

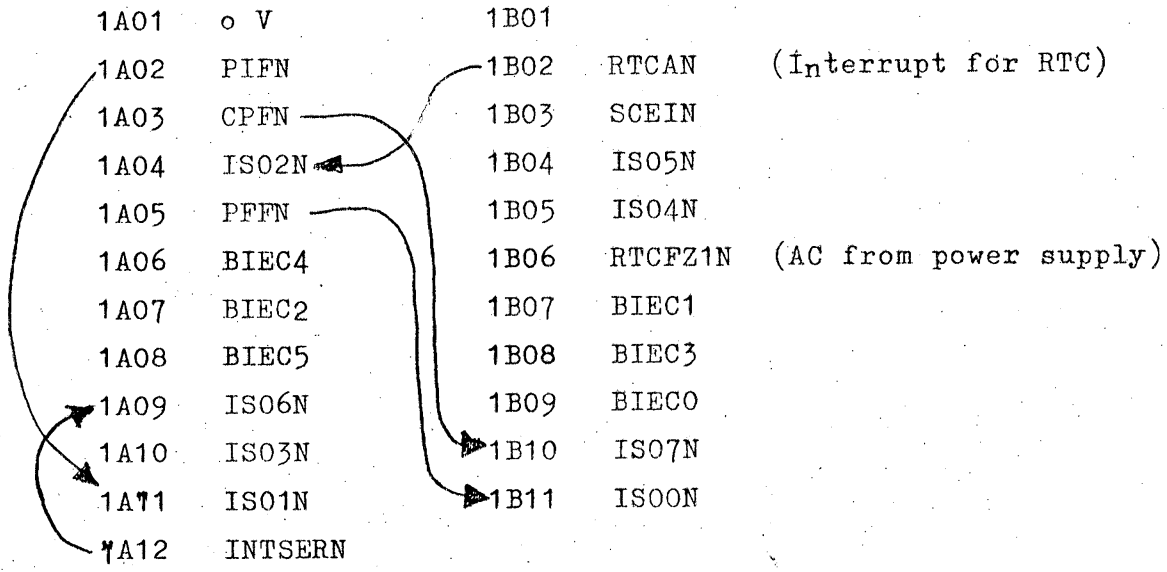
The P852-856-857 use the GP Bus. Part of this bus is the Interrupt Bus. This bus uses the 6 BIEC lines (BIEC= Bus Interrupt EnCoder)
This makes it possible to use jumpers on the control unit board to set the level of the interrupt. Level 00 has the highest priority, level 63 the lowest. This lowest level is obtained by master clear, it is the level of the CPU. The lowest hardware level is therefore 62.
The standard software is still based on the P855 structure that used only hardware levels 0--47. 48--62 were used as software levels there and still in our P852-856-857. That will give confusion if we really would have more than 48 devices that could cause interrupt!
The integral V 24 controller on the P856-857 CPU will give a special signal when "break" is pressed on the operators keyboard. This signal is OR-ed with the standard control panel interrupt. Very convenient if the operators machine is not standing beside the computer cabinet.
To understand the interrupt bus it is neccessary to remember that it is a two way bus. The interrupt encoders on the boards can not only send a signal on the bus, but also receive a (blocking) signal from other control units. Page 2-41 of the P800 Interface and Installation Manual shows a good example of an interrupt encoder.
Power Failure, RTC, integral V24 etc. are "internal" interrupts, They are generated by circuitry on, or closely connected to, the CPU board. They use levels 0--7. Unused levels can be selected on the controller boards. In any case the internal interrupt has higher priority, even if the external source generates a lower number.
Any fault condition in the Floating Point Processor will generate a FPP interrupt. To this an SST 0 is the answer. The status bits tell us what the error is.
If (A15) < /100 stack overflow interrupt is generated. In the P855 we got the stack overflow indication only if (A15)=/100.

The hardware that looks at the 6 BIEC lines is on page 2-70 of the P800 Service Manual. You find there that the internal interrupts(see below) use a priority encoder. Following this is a selector that looks only at the priority encoder if any of the internal interrupts is active. Otherwise it looks always at the BIEC lines. So, the internal interrupts always have higher priority than the external lines, whatever their level may be! Of course this is only meaningful if two interrupts occur at the same moment. Once they are accepted and running as a software program they work in the normal way by comparing the level of the program with the level of the new interrupt.

Internal Interrupt	Level
Power Failure	0
Auto Restart	0
LKM	1
Stack Overflow	1
RTC	2
Operators TY	6
Control Panel	7

This is the standard way of connecting the levels, it may be changed. It is shown on page 4-8, Service Manual. This copy here is without printing errors; and it shows the jumper wires.

Table 4-2 CPU-A Connector 1 (Interrupt & V24)



All CT100 signals are for the V24 connection.

The MCU3

The MCU3 (multiple Control Unit, for three devices) is for all Belier machines. It handles one V24 device (See V24 chapter),a reader and a punch.

Logically these devices are very similar to the P855M control units, but there three separate boards were used. The (Facit) punch controller is slighly modified: it senses if the punch unit is sending 6 V., to check if it is turned on.

All devices can have any address and any interrupt level, they can be operated by the IOP. Standard is to have them on the programmed channel.

On the board are two things that look like power transistors. This are two integrated circuit voltage regulators, one for +, the other for - They are only used by the V 24 circuits. (V24 prescribes levels of + or - 3 V. on the lines). The regulators are of different type, dont mix them up if you ever have to replace them!

The MCU2

The MCU2 handles one line printer and one card reader. If only one of the devices is needed, the same printed cicuit board is used. About half of the integrated cicuits are left out, then.

All other devices (Disc, Tape, Cassette) use a single board, single function controller. In this case more than one device can be connected to one controller.

V 24

The Comité Consultatif Internationale de Telephonie et Telegraphie (CCITT) Has published recommendations for connections. One chapter of this book is chapter V 24, it tells us how to connect Data Terminals to Data Communication Equipment. Because many Data Terminals are available it is nice to be able to connect them to the computer. The computer takes the place of the Data Communications Equipment. It must be able to handle the signals as described in chapter V 24. The unit that does this is very often called the V 24 Interface. You will understand that this name is not quite correct, but everybody uses it that way.

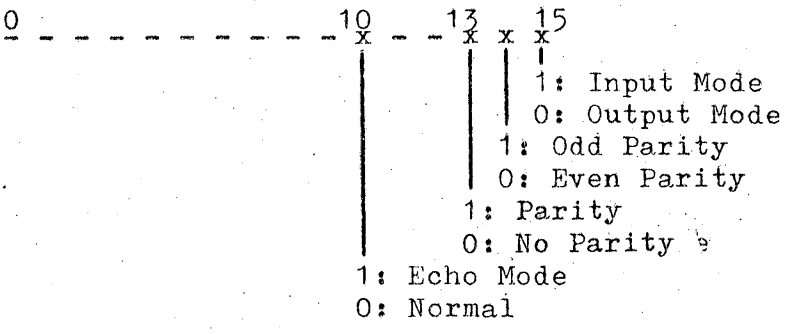
In the P800M two V 24 interfaces are present. One is part of the MCU3. This is a control unit for three devices, PTR, PTP and V24. The other is an integral part of the P856-P857 central processor. The difference is in the number of signals they handle and in the speed range of the units. See the table:

MCU3	856 857	PER 3100	V24 nr:	V24 Signal name	P800M---V 24 Device
0 V	x	x	101	Protective ground	⟷
SCREEN	x	x	102	Signal ground	⟷
TTD	x	x	103	Transmitted data	←
RRD	x	x	104	Received data	→
RTS			105	Request to send	←
CTS	x		106	Ready for sending	→
OPE	x	x	107	Data set ready	→
CNS	x	x	108/2	Data terminal ready	←
VAL	x		109	Receive line level	→
TRE			114/115	Transmit/receive clock	→
RFR	x	x	133	Ready for receiving	←

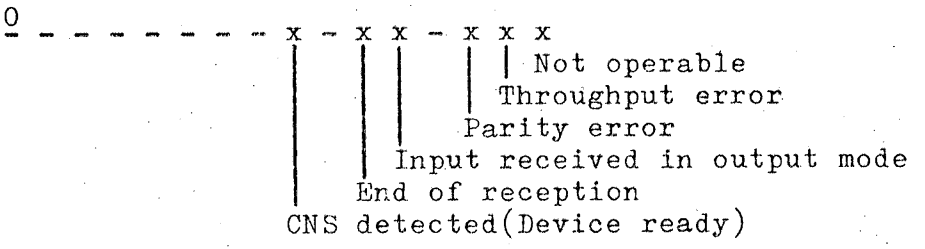
In the V 24 recommendation 35 lines are described. Only the ones mentioned here are used by us. The first column is the longest, this is the MCU3. There any V24 device can be connected. The integral P856-857 V24 controller is mainly intended for operators typewriters. A famous typewriter, The PER 3100 does not even need all these signals.

Note that the integral P852 control unit is a 20 mA current loop device. It handles the standard ASR33 at 10 char/sec. (110 Baud)

Some special I.O. instructions are used with V24 devices.
The register specified in the CIO Start instruction should contain:



The CIO Halt is normal, except for P855: If bit 8=1: CIO Off-Line. After stopping the device SST must be done, as usual, but not after a CIO Off-Line!
The status word received by the SST is this:



The integral V24 unit on the P856-P857 CPU board is described in section 6 of the Service Manual. It does not mention the setting of Y6 by D10 on page 6-6. This gives echo-mode like with the other V24. The parity, is selected by jumpers. This is shown on 4-12, or on 5-18 of the Interface and Installation Manual.

The UART generates a signal RFE if a stopbit error is detected. A stopbit error occurs when the break-key on the operators keyboard is depressed. Flip-flop AB7 (Page 6-6) is set and generates BRGFN. This signal goes to page 2-68 just beside signal CPINT. CPINT=Control Panel Interrupt, BRGFN does the same, so: Break-key=Interrupt Button.
On the PER3100 you will have to add something to make the Break-key operati

The PER3100 sometimes shows a peculiar error. On carriage return the print head moves the first few centimetres at normal speed and then continues very slowly. This seems to be an electronics problem, it is purely mechanical however. By cleaning and oiling the guide bars for the printhead it can be repaired. It is caused by a sensing circuit for the stepping motor current. If this is too high, the electronics switch over to a low speed. The high current is caused by the drag on the print head being too high.

The P851 has a SCUZ card (Serial Controller for the Z bus)
This can be used as V-24 Control Unit, or as 20 mA current loop Control Unit
Parity is programmed, echo-mode is obtained by a jumper.
The status word is funny:

```
0 - - - - - - - - - 8 - - - - 13 x x 15
                    x           x x x
input received |                | | not operable
in output mode -                | throughput error
                                parity error
```

The bits 9--12 contain any data that happens to be present in the UART receiver buffer register. After the SST an AND with constant /87 would clear the meaningless bits.

The jumpers on the board can connect 0--1 or 0--2
The speed is selected by jumpers CPR and CPT for receive and transmit.

Jumper	1	2	3	4
Speed:				
50	2	2	2	2
75	2	2	2	1
110	2	2	1	2
134.5	2	2	1	1
150	2	1	2	2
300	2	1	2	1
600	2	1	1	2
1200	2	1	1	1
2000	1	2	2	1
2400	1	2	1	2
3600	1	2	1	1
4800	1	1	2	2
7200	1	1	2	1
9600	1	1	1	2
19200	1	1	1	1

Device Address: put jumper on "1" for all zero bits, on "2" for all one bits. Bit 5 is the least significant bit.

	pos. 1	pos. 2
SB1	2 stop bits	1 stop bit
WL1	8 data bits	7 data bits
BD1	yes	no
(BD= input during output detection)		
ECHO	yes	no
CL1	Current Loop	V-24
CL2	2 wire	4 wire

OSC, EXGEN, TCP and RCP are only removed for test purposes. Normally they must be there.

If the keyboard Break-Key should work as control panel interrupt make the following connections on the P851 backpanel:
1A04(SCUZ)---2 11(P851) and 1C04(SCUZ)---2A09(P851)

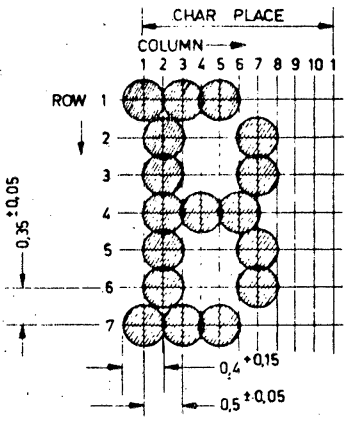
The 2230/2260 Line Printer.

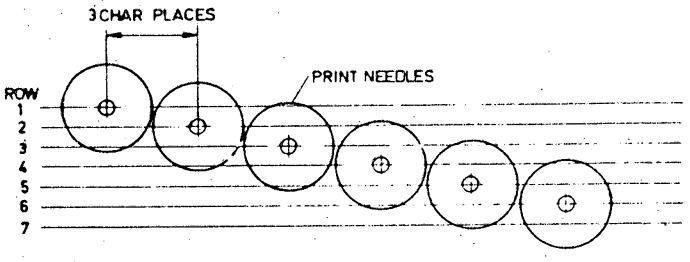
For P800M-S&I equipment the Data Products line printer ~~is~~ was more or less standard. The character set is identical to the ASR (ASCII)
136 ch/line, 300 lines/minute. Only 68 hammers are mounted, the hammerbank has to move from one position to the other to print all characters.
A faster model uses a fixed hammerbank with 136 hammers. It has twice the speed of the 2230, this is the 2260
The bits are numbered from right to left, 1 to 8 (Standard for ASCII)
If bit 8 =0, it is considered to be ASCII. If bit 8 =1 it is a Paper Feed Command. Some Paper Feed Codes are illegal, the line printer goes in the "Alarm" state if this code is received.

8	7	6	5	4	3	2	1	
0	x	x	x	x	x	x	x	ASCII
1	0	0	0	x	x	x	x	Tape Channel (1--12)
1	0	0	1	x	x	x	x	Nr. of lines (1--15)
0	0	0	0	1	0	1	0	Line Feed (ASCII /A)
0	0	0	0	1	1	0	0	Form Feed (ASCII /C)
0	0	0	0	1	1	0	1	Carr Retn (ASCII /D)

The little tape that controls the paper feed has 12 channels. In most of our applications only the top-of-form channel is used. This is channel nr. 1. Only one hole may be punched in this cannel. The length of the tape loop is related to the paper size. Number of sprockets in the tape = number of lines on one page.
All paper feed commands cause the printing of all buffer data. If the buffer is empty, only the paper feed is done. Sending a carriage return only at the end of a line makes it possible to print a line on top of another line. Sending only a Line Feed has the same effect as sending both a Line Feed and a Carriage Return.

The X1425 is nowadays (1979) the standard line printer for S&I. It is completely a Philips product and therefore cheaper (for us) than the 2230/2260. It uses the matrix principle to print a character. It can print 400 lines of 132 characters per minute. It uses only 50 print-needles to achieve this result. This is done by moving the whole bank of needles left and right over a distance of three characters. The X1425 (or the X1415, 200 lines per minute) is plug-compatible with the 2230/2260.
As an example of the printing quality see the section on Integrated Circuits in this book.



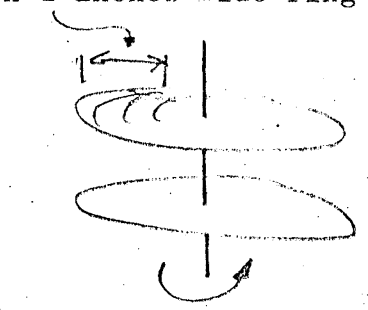


Careful with the needles! Do not clean them with any solvent, this will only wash the ink into the inner parts. The steel needle slides in a ruby, it is operated by a coil with cannot be repaired. The only way to clean the needles is to run the printer, in test-mode, without the ribbon. Ink deposits should go on the paper, then.
A needle which got stuck can be "helped". Dismount it, connect it to a power supply and apply a voltage that just moves the needle out. Now you may need a bit of alcohol... (for the needle). Now quickly connect and disconnect the power supply, until the needle appears to move freely. Then re-install it in the printer.
The easy way is, use a new needle...

the X1215 Disc.

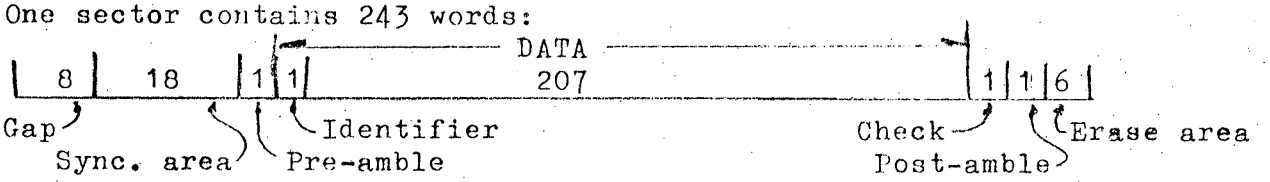
The X1215 is a Philips made disc drive. One machine contains two discs on one spindle. The lower disc is fixed to the drive, the upper one is housed in a removable cartridge.

A 2 inches wide ring on each disc surface contains 200 concentric tracks of serial information. On each side of each disc is a moveable read-write-erase head. At the bottom of each disc is a electro-mechanical sector sensing device. The discs are divided in 16 sectors. On one disc are in total 2x200x16=6400 sectors. In one sector go some 200 words, that means 200x6400=1280000 words on a disc. About 2.5 M words per machine!



2400 rpm
1 revolution=25 mS.

One sector contains 243 words:

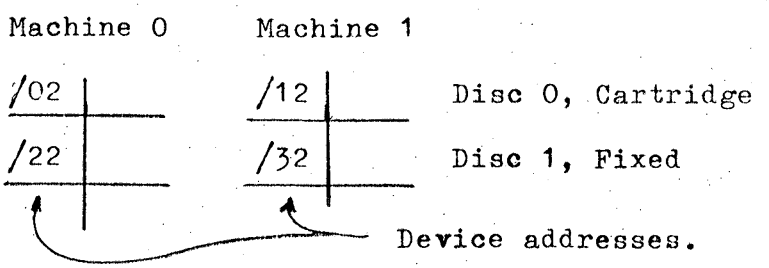


The standard P800M software uses 205 (=/CD) words per sector. The erase area is then 9 words. The identifier is one word that is written by the PREMARK program. This program is necessary to initialise new discs. The identifier only contains the track number. It is used, by the software, as a check on the seek operation. Only the DATA part can be altered by the program, the rest is only for checking and synchronization

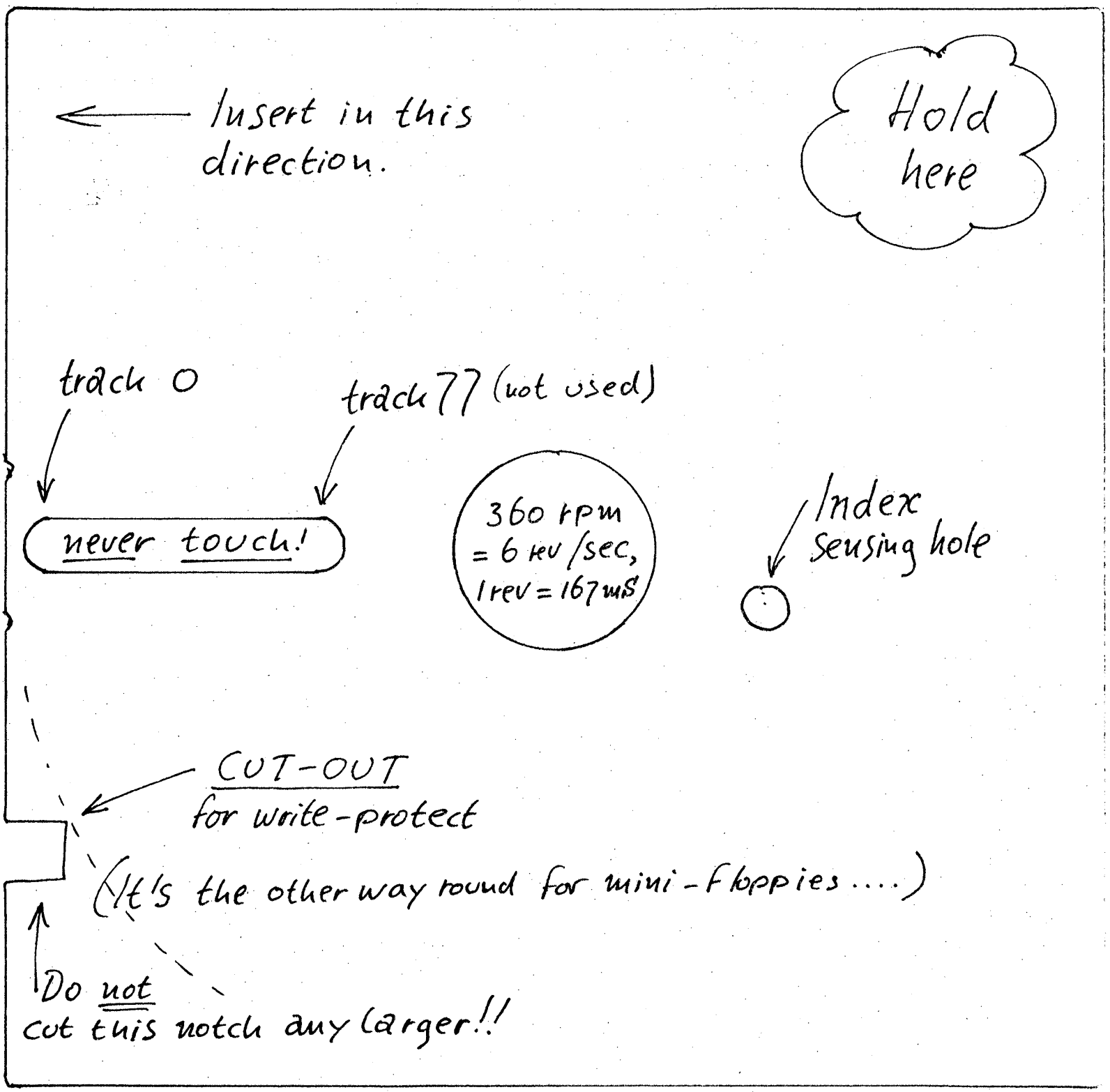
One track takes 25 mS, there are 16x243 words on a track, so the time per word is 6.5 µS. Gap + Erase area + Post-amble is 15 words, we have 100 µS between sectors. This is too short for the standard software to read or write two sectors consecutively. For this reason logical sector numbers are used. These logical numbers skip two physical sectors, so now there is plenty of time.

Physical number:	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Logical, upper :	0	11	6	1	12	7	2	13	8	3	14	9	4	15	10	5
Logical, lower :	16	27	22	17	28	23	18	29	24	19	30	25	20	31	26	21

Addressing may cause confusion:



Floppy Disc.

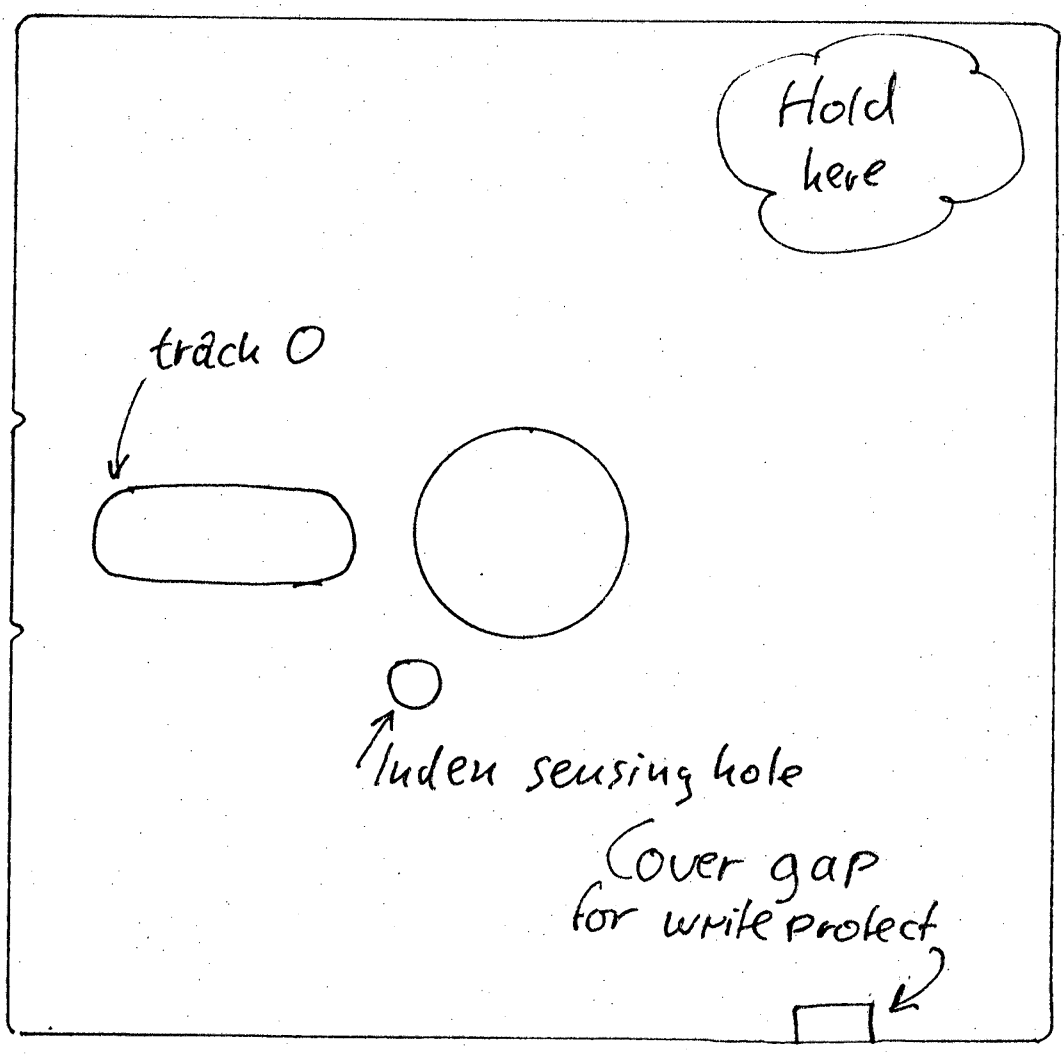


The floppies are pre-recorded (on a special machine) to have 26 records on one track. 26x77=2002, so the record numbers run from 0 to 2001. Each record contains 64 words that are accessible to the user.(or the pre-mark program). Because the control unit is word-oriented, better not talk about bytes. In total there are 128128 words.

-Never use a bulk-eraser, you can't get the soft-sectors back on again-

A damaged floppy could be reversed in it's envelope, if you are a good surgeon. The problem is, that the pre-recorded record numbers are not present.
There does exist a floppy with two index sensing holes in the envelope (and only one in the disc itself, of course) This floppy can be used on both sides. As reversing the floppy is a manual operation it only halves the cost of the floppies, and these are cheap already.
In the future a drive with two write/read heads may be used. This will really double the available capacity, ass seen by the machine.

In the PM4400 mini floppies are used. These are similar devices, but of much smaller size



In the near future (now is: 1979) double sided-double density floppies will be used. Later: double number of tracks.
The mini floppy is not (yet) a P800 standard device.

The Input-Output Processor

The IOP enables very fast input-output. It is rather different from the P855M multiplex. It has its own registers, therefore it can work at almost memory speed. In the P851M one IOP is possible, in the Beliers you can have more. With great trouble I can think of an example with 2 IOP's, I think the maximum of 8 is never used.

Each IOP has 8 channels, each channel can handle 1 control unit.

The Belier Bus makes it easy to design an IOP. There is no great need for DMA (Direct Memory Access) The DMA speed = IOP speed.

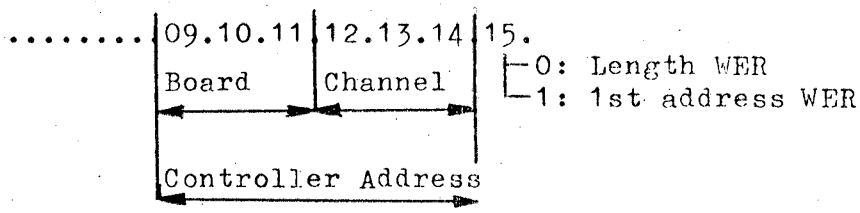
But some devices like to have their own high speed control unit. If the part that is normally in the IOP is included in the control unit, we speak of DMA. (Again I can't think of a standard DMA example)

There is a very close relationship between device address and channel number, it must be the same. For each channel there are two registers in the IOP. One contains the first address of the buffer, and the other the length of the buffer that is used. These registers are loaded with WER instructions before the data transfer is started. The address for the first WER is twice the channel number. For the second WER it is twice the channel number+1. See the little program here, for driving the line printer on the IOP:

```
        0007      LP        EQU       7                      L.P. ADDRESS
                  *PRINT SUB-ROUTINE:
0000    9220      PRINT     ADKL      A2,/4000               LENGTH,OUTPUT
0002    4000
0004    720E                WER       A2,LP+LP
0006    710F                WER       A1,LP+LP+1             START ADDRESS
0008    41C7                CIO       A1,1,LP
000A    49C7                SST       A1,LP
000C    5C04                RB(NA)    *-2                    WAIT FOR I.O.P.
000E    1100                ADK       A1,0                   ERROR?
0010    5C04                RB(NZ)    *-2
0012    F03A                RTN       A14
```

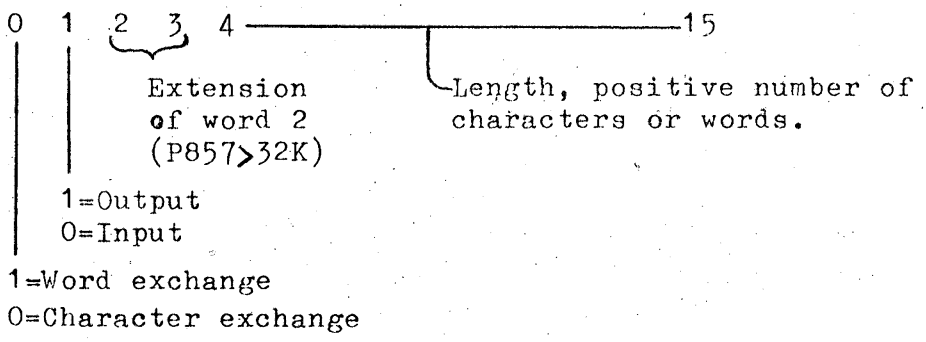
Here the IOP nr. 0 is used.

There seems to be a problem with the **controllers** that handle more than one machine. (Disc, Tape, Cassette) In the disc controller the addresses /02, /12, /22 and /32 can be used. It appears now that we need four IOP channels, that are not even on the same IOP board. This is not true: the controller has the address /2, the leading digit points to a single machine, connected to that controller. So the WER's would have the number 4 or 5 for all machines, the difference would only be in the CIO or SST instructions.
So the IOP makes it impossible to have two discs going at the same time. No problem, it's not possible with the controller anyway.



A small problem is channel nr. 0. The use of device address 0 is restricted; SST 0 is used to check the FPP status. (FPP= Floating Point Processor, P857M only.)
All devices can work with the IOP, except the integral control unit on the CPU board and some MIOS controllers. It is standard however, to use only the faster devices with the IOP
To make this story complete we copy part of page 65 of the P800M Programmer's Reference Data booklet.

Word format for 1st WER:



Word format for 2nd WER: just the start address of the memory area.

The Trap

If a CPU tries to do an instruction that is not in its repertoire, trap action takes place. Maybe the instruction code does not exist, then it is treated as illegal. Sometimes it is an optional instruction and you did not buy the option. Then it can be simulated by the software. The price you have to pay then is the much longer execution time, otherwise it has exactly the same result.

In the P852 are no optional instructions, but you can trap here the P855 instructions like ML, MS, MU, DV and double shifts.

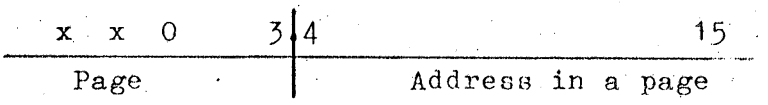
In the P857 the FPP and MMU are optional. The Floating Point Processor instructions can be simulated by the software, but for the MMU this is not possible in a practical way. You can not simulate memory that you don't have...

Not-wired instructions cause a LKM interrupt in the P855. Here it is quite different. Trap action will also take place if the program runs under Inhibit. Similar to interrupt, P and PSW are stored in the stack. The program continues on the indirect address /7E. (Highest hardware level address= /7C). Difference with interrupt is that the program level register is not changed. At the end of the trap handling program the software returns to the User program and enables interrupt. This may cause confusion: If your program ran under Inhibit, it will be Enabled now! Of course normal programs always run in the Enable mode.

In the P852-856-857 the LKM interrupt is only caused by LKM or Stack Overflow. It has nothing to do with Trap.

The Memory Management Unit.

In the P857 the MMU is available as an option. It gives us the possibility to use more than 32K, with a maximum of 128 K (1K=1024) Also memory protection is possible, with an organization rather different from the P855. With the MMU the memory is split up in pages



The 6 page bits give a maximum of 64 pages. The 12 remaining bits give us an address in a 2K page.

The two x bits can be easily obtained from the IOP. (see page 12)

The MMU itself does this in another way: the bits 0--3 are used to point to a word in a 16 word table. The 6 leading bits of this word indicate a page number. The 10 remaining bits are used for various special purposes. One of these is memory protection.

The MMU is completely and automatically served by the MAM, the Multi-Application Monitor. This monitor allows multi-programming. Each program gets its own page number table. This contains only 16 page numbers, so no single program can be greater than 32K.

There is a special group of instructions to handle the MMU hardware. Any error condition in the MMU generates an interrupt. The interrupt handler can detect by doing SST 0 what type of error this is.

I mentioned already that the Trap cannot simulate a non-existing MMU option. The result of the MMU instruction will be insignificant in this case.

The Instruction set.

Only the differences between the CPU's are pointed out here.

P852 has no multiple register instructions, no multiply/divide and no double shift instructions. Both P851 and P852 have no memory protection hardware, so also no SMD instruction.
All multiply/divide instructions are optional on the P855.
The "Executu" and "External Register" instructions are not available in the P855. They cannot be simulated by the standard monitors either.
"WIM" and "RIL" are instructions only for the different P855 interrupt structure. "WMP" and "WM2" are only for the P855 Memory Protection Option.
"RCA" is only for the P855 DMA Option.
The P857 has a set of "Move Table" instructions. The P857 also has the possibility of MMU and FPP options. There are special load/store instructio to enable working in store greater than 32k. With the FPP come a group of instructions to load, store or convert floating point numbers, and to do the 4 main arithmetic operations.

To get an idea of the speed some instruction times are compared:

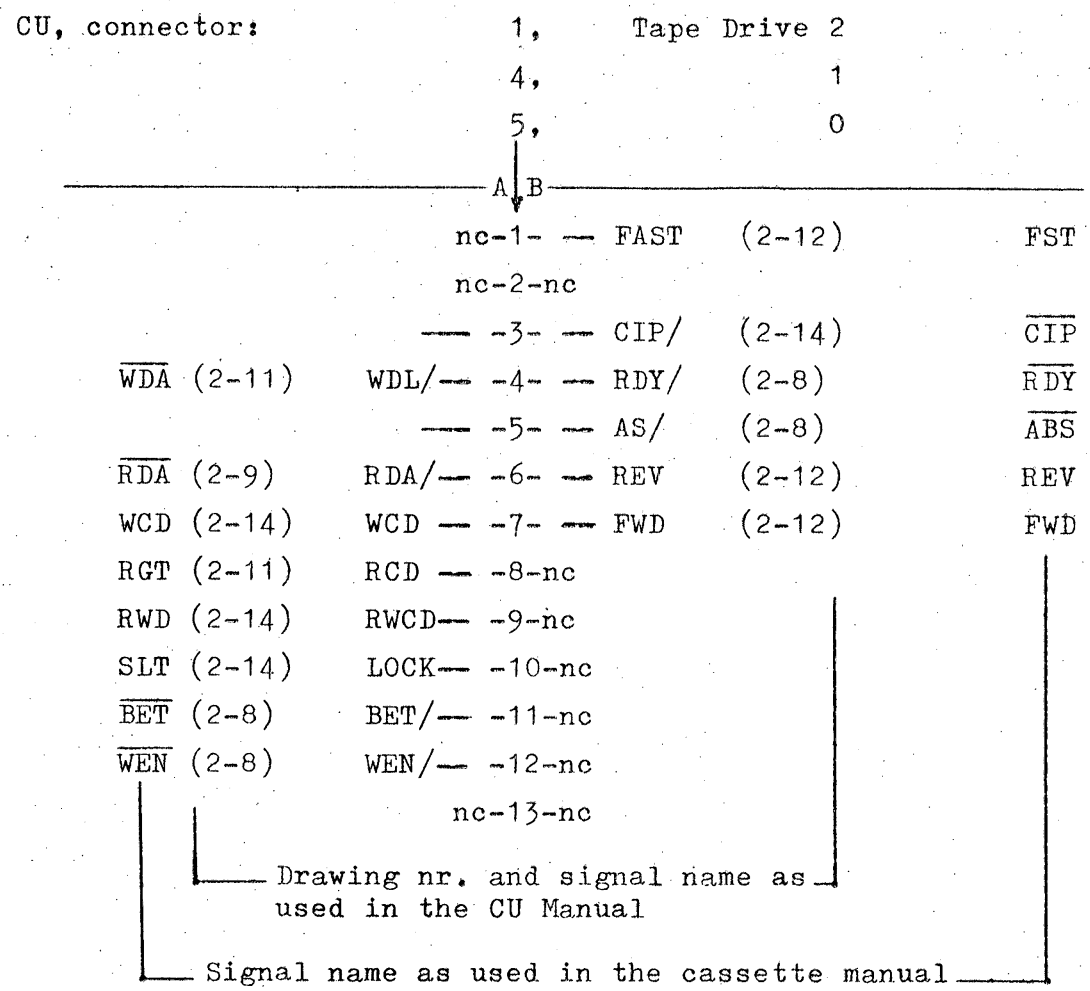
	P851	P852	P855	P856/P857	
Memory cycle:	750	1200	840	700	1200 nS
LDK :	3.1	1.6	1.56	0.9	1.3 uS
AD* A1,MM :	12.36	6.2	3.36	2.9	5.1 uS
CF A14,SR :	11.7	4.7	5.64	2.7	3.8 uS
	27.16	12.5	10.56	6.5	10.2 uS

The very high speed obtained with the P856/P857 700nS memory is only possible if maximum "interleaving" takes place. A practical value will be somewhere around 8uS.
In programs where a lot of multiply/divide is done, the P851 may be faster than the P852. In FORTRAN programs, the P857+FPP will be much faster than any other P800.
In general you could say that P852 has 2x the speed of P851, and P855/6/7 have 3x the speed of P851.

Most of the monitor programs can be organized in such a way that they simulate by software the instructions that are missing in the repertoire of a certain CPU. Of course this solution costs a lot of time and memory. It is not possible for all types of instuction.

The Cassette.

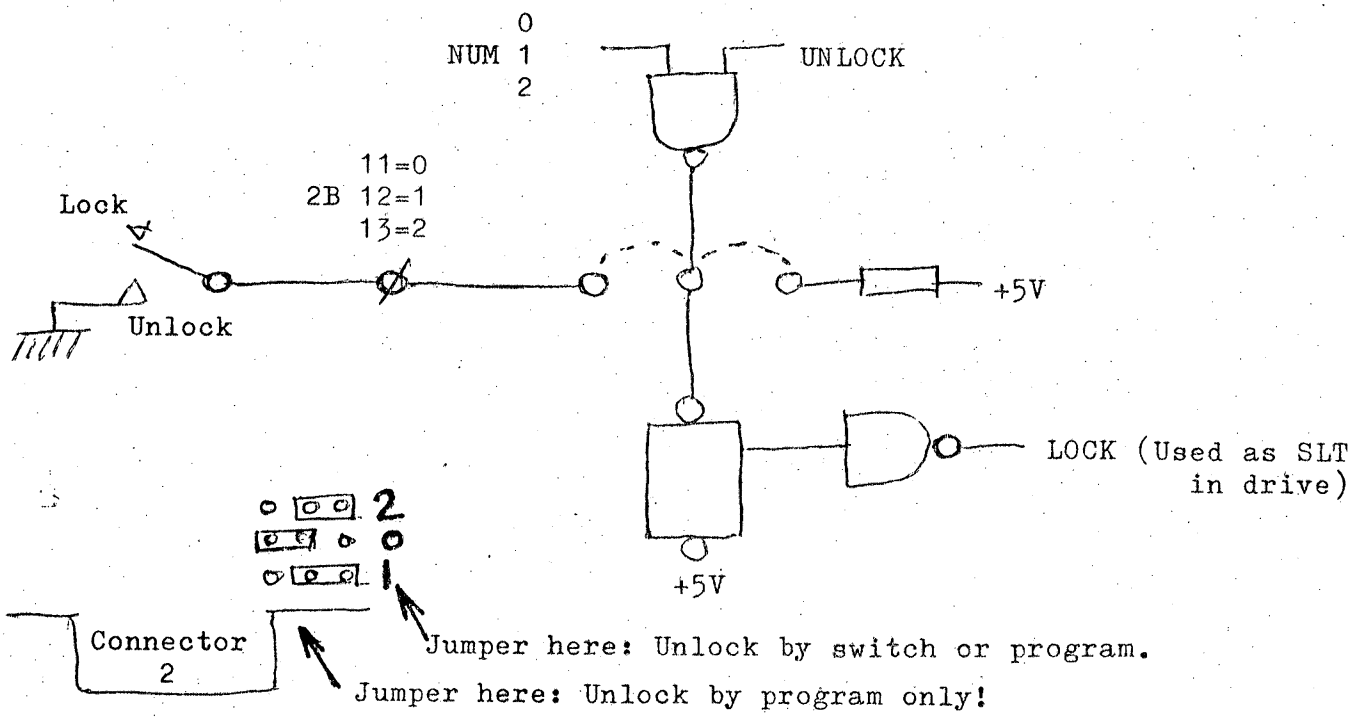
If you pronounce K7 in French, you will understand why this is used as name for the Cassette system. In the following pages you will find some information that is not so easy to extract from the Cassette Tape Control Unit Manual (5122 991 12461)

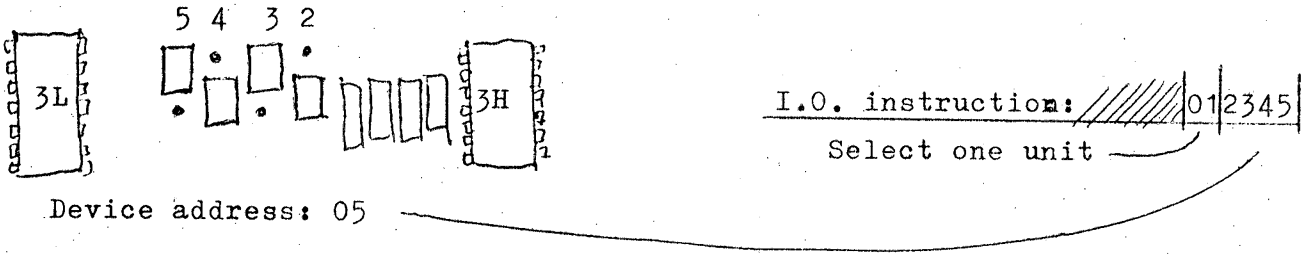


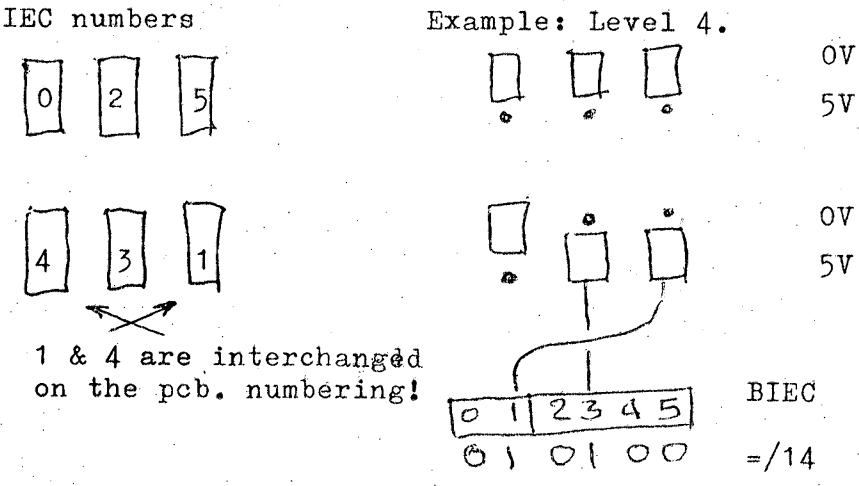
ABS	A or B side	RDY	Ready
BET	Begin/End of tape	RGT	Read gate
CIP	Cassette in position	RWD	Rewind
FST	Fast winding	SLT	Select
FWD	Forward	WCD	Write command
LCK	Cassette locked	WDA	Write data
REV	Reverse	WEN	Write enable plug
RDA	Read data		

Note: Cassette SLT= CU LOCK!

On cassette drive TT board SLT jumper should be installed!







Address /5 and level /14 are standard system values!

The blocklength used by the standard software is $\frac{1}{4}$K, or 256 characters. To this must be added 4 characters for synchronization and checking About 1000 of these blocks go on a tape of 86 metres. One block is about 86 millimetres. As the density is 800 bits/inch there go 100 characters/inch. So, one block=260 characters=2.6 inch=62.5 mm. Interrecord gap must be 86-62.5=23.5 mm.

The 2.8 million bits/track, as mentioned in the manual is only possible if you write extremely long records. (Possible with the hardware!)
In practice you could get about 2 million bits on a track.
In most cases only the first few metres of a cassette are used, otherwise you lose too much time just winding your tapes.

The time to write one block can be calculated:
Speed: 750 characters/second
Start time: 20 mS
Stop time: 20 mS
so, for one block: $\frac{260}{750} + \frac{20+20}{1000} = 385$ mS

1000 blocks take 385 S, = $6\frac{3}{4}$ minutes.

In practice this is about 8 minutes, so the speed must be lower than stated

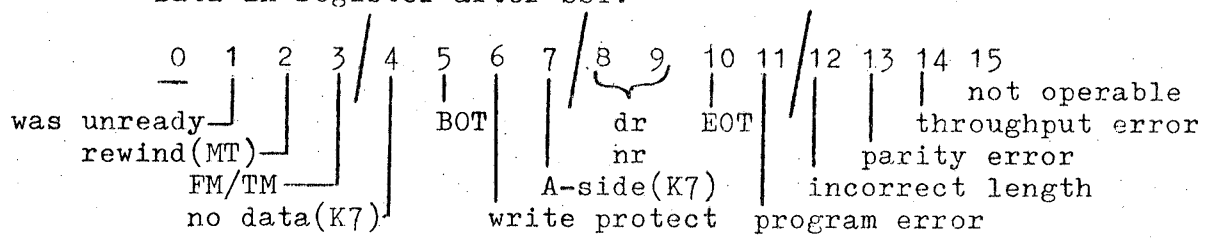
As the control unit of the cassette is housed in the shelf that also contains the drives, it is possible to use them on a P851+IOTZ

Cassette & Mag. Tape programming

Register during CIO Start:

MT	K7	
/20	0	Unlock/Off line
/19	1	Erase
/04	2	Backspace one block
/00	-	Forward space one block
/03	5	Write a block
/02	7	Read a block
/24	8	Rewind
/11	9	Write File Mark/Tape Mark
/14	A	Search FM/TM backward
/10	B	Search FM/TM forward

Data in register after SST:

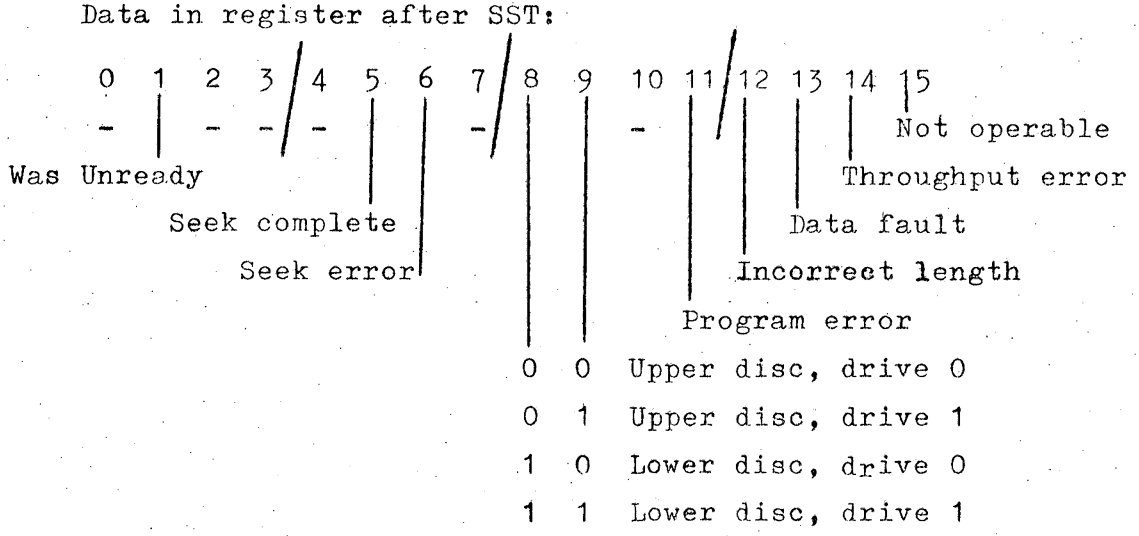


Disc programming:

Register during CIO Start:

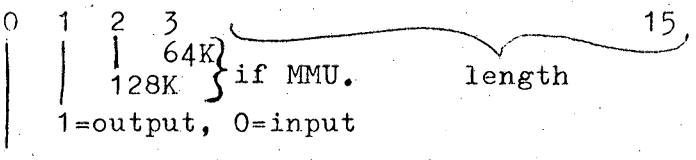
3	Seek Zero
8xT+2	Seek track T
4xS+1	Write sector S
4xS	Read sector S

Data in register after SST:



I.O.P programming: Channel nr= Device address, reg 1=2xch. nr., reg 2=2x ch. nr. +1. (reg 2)= starting address.

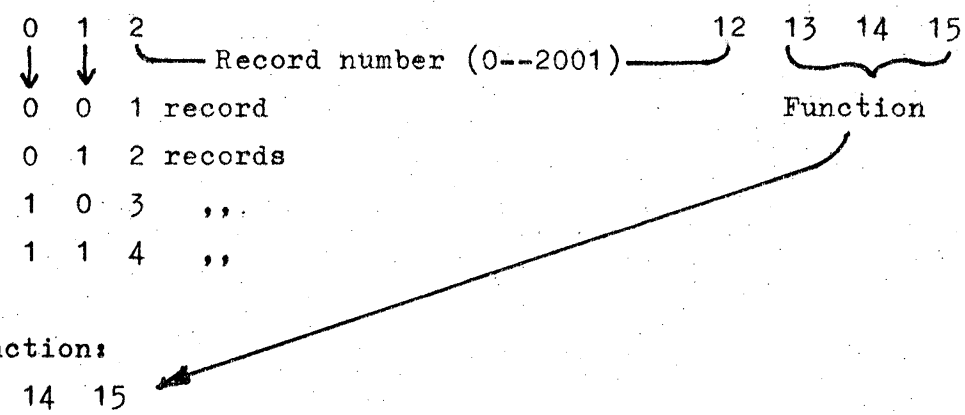
reg 1:



Floppy Disc Programming.

The C.U. contains 4 registers of 64 words. This means that 4 records may be read or written consecutively.

Register specified during CIO Start:



Function:

13	14	15	
0	0	0	Read
0	0	1	Write
0	1	0	Write with Deleted Data Address Mark
0	1	1	Search Key
1	0	0	Read record with Key (Only after "search key")
1	0	1	Write & Verify (Write, then read and compare with buffer)
1	1	0	Write with Deleted Data Address Mark and Verify.
1	1	1	Search Key with Mask (Ignore spaces)

When doing "Search Key" bits 5-12 must contain the length of the key in characters. The "Key" should be send to the buffer register by OTR's or by I.O.P. cycles. The key should contain the number of characters as mentioned before. It should be followed by two words containing the first and the last record number to be searched. After a succesful search a "Read with Key" must be done. The data from the record is preceded by the record number, so 65 words must be transferred. A one word transfer (Only the record number) is possible, then the "incorrect length" must be ignored

Status bits:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
-	Ready after not ready	Key not found	-	Deleted Data Address Mark	Record not found	Write protected	-	Drive number	Drive number	Retry	Program error	Incorrect length	Data Fault	-	Not operable

Floppy search key.

Assume you want to find a record beginning with the letters 'AB' from record 300 upto 2000. The data that must be sent to the floppy control unit buffer is then:

```
PTRN  DATA  'AB',300,2000
```

The following program is necessary:

```
      INH
      LDKL A1,PTRN
      WER  A1,7      assuming a CU address of /03
      LDKL A1,/C003  output of three words by I.O.P.
      WER  A1,6
      LDK  A1,/13    bit 11-12= 1 0 =2=keylength
      CIO  A1,1,3
      SST  A1,3
      RB(4) X-2
      ENB
```

If the program reaches this point, check if the status is 0.
If bit 2 is on (status=/2000) then the key can't be found
If all is fine, then:

```
      LDKL A1,/8001  input of one word
      WER  A1,6
      LDKL A1,RECNO
      WER  A1,7
      INH
      LDK  A1,4      read-with-key
      CIO  A1,1,3
      SST  A1,3
      RB(4) X-2
      ENB
```

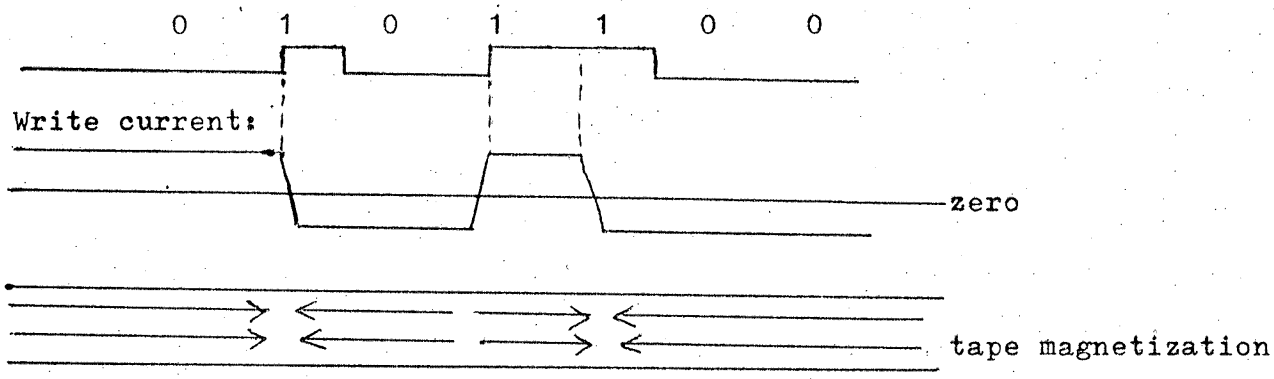
If, at this point, the status is /0008, it's O.K., we should have incorrect length now because we read only one word. Now RECNO will contain the record number. It is also possible to specify a length of 65. Then organize the memory so, that BUF follows RECNO, if BUF is the area where the data should go:

```
RECNO RES 1
BUF   RES 64   <-- Nothing in between!
```

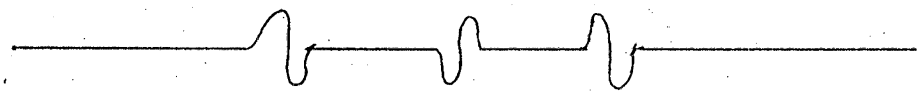
NRZ1 = Non-Return-to-Zero, change on 1

A method to put binary data on magnetic recording material.

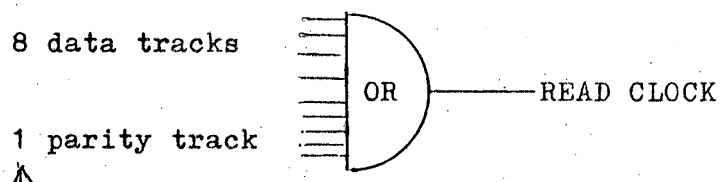
Data in one track:



Read pulses:



So, any pulse=1, no pulse=0!

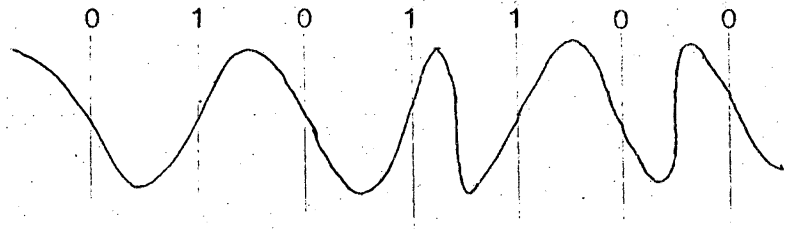


If odd parity is used, there is always at least one bit in one of the tracks, to generate a clock pulse.

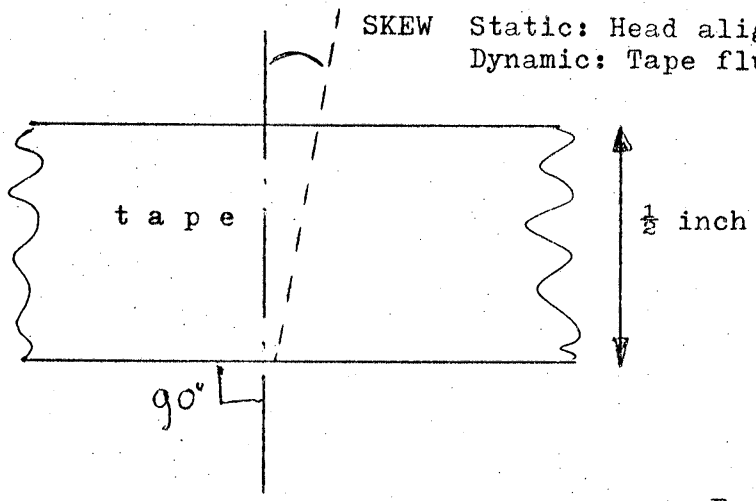
PE = Phase Encoding

More modern than NRZ1, needs more complicated electronics. Nowadays 1 I.C. to do Zero-Cross detection in a channel.

Data in one track:



The direction of a zero crossing determines if we read a one or a zero. Each data pulse triggers a one-shot (one per channel!) for a certain time, so you don't see the insignificant crossings. In this way synchronization per track is possible

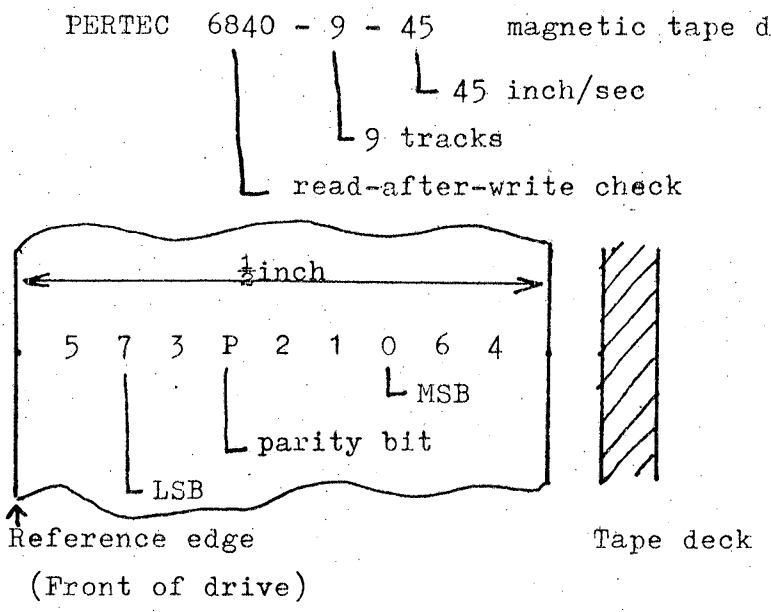


If for each track a shift register of say 5 positions is used electronic skew correcti of up to 5 frames is possible, because each track generates its own clock pulses.

For this reason higher densities ar possible with PE than with NRZ1.



If you have this much skew in a NRZ1 system you are already reading bits from differen frames!



HIGH DENSITY is an indicator only, on 9-ch, NRZ1 decks the density is always 800 bits/inch

Here is a test-program:

```
2000 20BF INH
2002 01FF LDK     A1,/FF
2004 8220 LDKL    A2,/1000
2006 1000
2008 8149 ST      A1,/4000,A2      } fill a buffer, 00,FF,00,FF, etc.
200A 4000
200C 1A02 SUK     A2,/02
200E 5C08 RB(4)   /2008
2010 0324 LDK     A3,/24
2012 F6A1 CF      A14,/3000
2014 3000
2016 8120 LDKL    A1,/FC18         =-1000
2018 FC18
201A 8141 ST      A1,/2100
201C 2100
201E 0303 LDK     A3,/03           }
2020 8420 LDKL    A4,/4FFF         }
2022 4FFF                          }
2024 8520 LDKL    A5,/4000         } write 1000 blocks
2026 4000                          }
2028 F6A1 CF      A14,/3000        }
202A 3000                          }
202C 9041 IM      /2100            }
202E 2100                          }
2030 5C14 RB(4)   /201E            }
2032 5F34 RB      /2000
```

the subroutine /3000 is shown on the next page.

ASHCAN was used to make this program.

This is a general-purpose magnetic tape routine.
Error messages include the I.O. status word, and the value of the P-counter of the calling program.
(A3) pattern used by CIO Start
(A4) word 1 for I.O.P.
(A5) word 2 for I.O.P.

```
3000 7408 WER     A4,/08
3002 7509 WER     A5,/09
3004 43C4 CIO     A3,1,/04
3006 542E RF(4)   /3036        if CIO not accepted
3008 49C4 SST     A1,/04
300A 5C04 RB(4)   /3008
300C 8204 LDR     A2,A1
300E 221F ANK     A2,/1F       check for error bits
3010 543E RF(4)   /3050
3012 F03A RTN     A14          no error: return to main program
          RES /0001
3016 2840 ENB
3018 0605 LDK     A6,/05
301A 2804 LKM                  print (A1)= I.O. status
301C 8620 LDKL    A6,/3100
301E 3100                      print: 'ST. (p)='
3020 2804 LKM
3022 815A LD      A1,/0004,A14
3024 0004
3026 1904 SUK     A1,/04       Get P-value from stack, to find location
3028 0605 LDK     A6,/05       of the CF that caused the error.
302A 2804 LKM
302C 0606 LDK     A6,/06
302E 2804 LKM
3030 20BF INH
3032 5F34 RB      /3000        try again
          RES /0001
3036 4384 CIO     A3,0,/04     here if CIO start was not accepted
3038 49C4 SST     A1,/04
303A 5C04 RB(4)   /3038
303C 2840 ENB
303E 810C LDR     A1,A3        print (A3)= CIO pattern
3040 0605 LDK     A6,/05
3042 2804 LKM
3044 8620 LDKL    A6,/3200     print: 'CIO N.A.'
3046 3200
3048 2804 LKM
304A 20BF INH
304C 5F4E RB      /3000        try again
          RES /0001
3050 2201 ANK     A2,/01
3052 5C54 RB(4)   /3000
3054 5F40 RB      /3016
```

If the error is: Not operable, then try again. Otherwise go to /3016

Print messages:

```
3100 5354 DATA 'ST. (P)='
3102 2E20
3104 2850
3106 293D
          RES /3200
3200 4349 DATA 'CIO N.A.',/DOA
3202 4F20
3204 4E2E
3206 412E
3208 0D0A
```

```
PAGE 1     DATE  76-05-21          IDENT     COPMPX


00000                              IDENT     COPMPX
00001                    *COPY-MULTIPLEX
00002          0001      PR        EQU       1
00003          0003      PP        EQU       3
00004  0000    20BF      START     INH
00005  0002    0164                LDK       A1,100
00006  0004    7102                WER       A1,2
00007  0006    8220                LDKL      A2,BUF
       0008    002C  R
00008  000A    7203                WER       A2,3
00009  000C    41C1                CIO       A1,1,PR
00010  000E    49C1                SST       A1,PR
00011  0010    5C04                RB(NA)    *-2
00012  0012    1100                ADK       A1,0
00013  0014    5002                RF(0)     *+4
00014  0016    207F                HLT
00015                    *ERROR, (A1)=STATUS
00016  0018    8120                LDKL      A1,100+/4000          OUTPUT BI
       001A    4064
00017  001C    7106                WER       A1,6
00018  001E    7207                WER       A2,7
00019  0020    41C3                CIO       A1,1,PP
00020  0022    49C3                SST       A1,PP
00021  0024    5C04                RB(NA)    *-2
00022  0026    1100                ADK       A1,0
00023  0028    582A                RB(0)     START
00024  002A    207F                HLT
00025                    *ERROR HALT FOR PP
00026  002C                BUF       RES       50
00027                               END       START
```

This is not a program for practical use, but it proves that reader and punch can be operated by the IOP. If you want to try this, change the device addresses as shown on line 2 and 3. Do not forget to change these addresses back again after running the program, or the standard software will not work!

```
PAGE 1     DATE  76-05-21          IDENT     POFATS


00000                              IDENT     POFATS
00001                    *POWER FAILURE TEST FOR P852.
00002          0000      START     EQU       *
00003  0000   8320                LDKL      A3,500              RTC TIMER
       0002   01F4                                              10 seconds
00004  0004   8120                LDKL      A1,POFA
       0006   0050  R
00005  0008   8141                ST        A1,/0               LEVEL 0
       000A   0000
00006  000C   8120                LDKL      A1,RTC
       000E   002A  R
00007  0010   8141                ST        A1,/4               LEVEL 2
       0012   0004
00008  0014   8120                LDKL      A1,ASR
       0016   003C  R
00009  0018   8141                ST        A1,12               LEVEL 6
       001A   000C
00010  001C   8120                LDKL      A1,BUTTON
       001E   004A  R
00011  0020   8141                ST        A1,14               LEVEL 7
       0022   000E
00012  0024   87A0                LDKL      A15,/110            STACK
       0026   0110
00013  0028   5F02                RB        *
00014                    *
00015  002A   20F7      RTC       RIT       /1B
00016  002C   1B01                SUK       A3,1                500-1-->A
00017  002E   5002                RF(0)     *+4
00018  0030   F03E                RTN       A15                 IF NOT ZE
00019  0032   8320                LDKL      A3,500                    ZERO
       0034   01F4
00020  0036   0400                LDK       A4,0                OUTPUT
00021  0038   44D0                CIO       A4,1,/10
00022  003A   F03E                RTN       A15
00023                    *
00024  003C   0407      ASR       LDK       A4,7
00025  003E   4410                CTR       A4,0,/10
00026  0040   5404                RF(NA)    SST
00027  0042   4490                CIO       A4,0,/10            STOP ASR
00028  0044   F03E                RTN       A15
00029                    *
00030  0046   49D0      SST       SST       A1,/10
00031  0048   F03E                RTN       A15
00032                    *
00033  004A   20DF      BUTTON    RIT       /F
00034  004C   0319                LDK       A3,25               1/2 SECO
00035  004E   F03E                RTN       A15
```

```
PAGE 2    DATE  76-05-21          IDENT    POFATS

00036                      *
00037  0050  8120          POFA     LDKL     A1,RESTAR
       0052  005A  R
00038  0054  8141                   ST       A1,/0          The re-start
       0056  0000                                                address.
00039  0058  207F                   HLT
00040                      *
00041  005A  20EF          RESTAR   RIT      /17
00042  005C  87A0                   LDKL     A15,SPST
       005E  0060  R
00043  0060  F03E                   RTN      A15                    TO THE MA
00044                      *
00045        0060          SPST     EQU      *-2
00046  0062  FC00                   DATA     /FC00,START
       0064  0000  R
00047                               END      START
```

This program rings the ASR bell every 10 seconds. When INT is pushed, the bell rings after one half second. When the panel switch is in the LOCK position, power failure interrupt will work. If the power is disconnected and then connected again, the program should continue.

SPST should contain: /FC40,START to set the ENB ff at RTN as this is not automatically set, in P851, P856 & P857.

```
PAGE 1     DATE  76-05-21          IDENT     LPTRY

00000                              IDENT     LPTRY
00001          0007        LP      EQU       7                        L.P. ADDRE
00002                      *PRINT SUB-ROUTINE:
00003   0000   9220        PRINT   ADKL      A2,/4000                 LENGTH,OU
        0002   4000
00004   0004   720E                WER       A2,LP+LP
00005   0006   710F                WER       A1,LP+LP+1               START ADD
00006   0008   41C7                CIO       A1,1,LP
00007   000A   49C7                SST       A1,LP
00008   000C   5C04                RB(NA)    *-2                      WAIT FOR
00009   000E   1100                ADK       A1,0                     ERROR?
00010   0010   5C04                RB(NZ)    *-2
00011   0012   F03A                RTN       A14
00012                      *
00013   0014   20BF        START   INH
00014   0016   8120                LDKL      A1,BUF1
        0018   004C   R
00015   001A   8220                LDKL      A2,BUF2-BUF1
        001C   0010
00016   001E   F6A1                CF        A14,PRINT
        0020   0000   R
00017   0022   8120                LDKL      A1,BUF2
        0024   005C   R
00018   0026   8220                LDKL      A2,BUF3-BUF2
        0028   0018
00019   002A   F6A1                CF        A14,PRINT
        002C   0000   R
00020   002E   8120                LDKL      A1,BUF3
        0030   0074   R
00021   0032   8220                LDKL      A2,BUF4-BUF3
        0034   000C
00022   0036   F6A1                CF        A14,PRINT
        0038   0000   R
00023   003A   8120                LDKL      A1,BUF4
        003C   0080   R
00024   003E   8220                LDKL      A2,END-BUF4
        0040   0014
00025   0042   F6A1                CF        A14,PRINT
        0044   0000   R
00026   0046   2840                ENB
00027   0048   2804                LKM
00028   004A   0003                DATA      3
00029                      *
00030   004C   4142        BUF1    DATA      'ABCDEF',/D0A,'GHIJKL',/D0A
        004E   4344
        0050   4546
        0052   0D0A
```

```
PAGE 2     DATE  76-05-21          IDENT     LPTRY

           0054  4748
           0056  494A
           0058  4B4C
           005A  0D0A
00031      005C  5858        BUF2      DATA      'XXXXXX',/200A,'YYYYYY',/200A
           005E  5858
           0060  5858
           0062  200A
           0064  5959
           0066  5959
           0068  5959
           006A  200A
00032      006C  5A5A                  DATA      'ZZZZZZ',/D0A
           006E  5A5A
           0070  5A5A
           0072  0D0A
00033      0074  4142        BUF3      DATA      'AB',/7F91,'CD',/7F92,'EF',/7F9
           0076  7F91
           0078  4344
           007A  7F92
           007C  4546
           007E  7F93
00034      0080  454E        BUF4      DATA      'END OF THIS1 TEST.'
           0082  4420
           0084  4F46
           0086  2054
           0088  4849
           008A  5354
           008C  2054
           008E  4553
           0090  542E
00035      0092  0D0A                  DATA      /D0A
00036            0094        END       EQU       *
00037                                  END       START
```

The main objective of this program is to try the various print commands. If, in the data stream, a /D or a /A appears the internal line printer buffer contents will be printed at once. The buffer will be cleared, and the next data will go to column 1 of the printer. If the code is /A line feed will be performed too.

Codes like /91, /92 or /93 will all cause printing, and the spacing of one, two or three lines. Maximum is /9F, for 15 lines.

```
PAGE 1    DATE  76-05-21          IDENT     DISKIO


00000                              IDENT     DISKIO
00001                    *TEST IOP+DISK SPEED
00002                    * (A5)=LOOPCOUNTER
00003                    *
00004   0000   20BF     START     INH
00005   0002   01CD               LDK       A1,/CD                    LENGTH IN
00006   0004   0304               LDK       A3,4
00007   0006   7104     TRY       WER       A1,4
00008   0008   8220               LDKL      A2,/4000
        000A   4000
00009   000C   7205               WER       A2,5
00010   000E   43C2               CIO       A3,1,2                    READ A SE
00011   0010   0500               LDK       A5,0
00012                    *
00013   0012   4AC2     SST       SST       A2,2                      ** 5.3 **
00014   0014   5414               RF(NA)    COUNT                     ** 1.8 **
00015   0016   221F               ANK       A2,/1F                    ONLY ERRO
00016   0018   5004               RF(Z)     *+6
00017   001A   207F               HLT                                 IF ERROR,
00018   001C   5F1E               RB        START
00019   001E   1304               ADK       A3,4                      NEXT XECT
00020   0020   EB20               CWK       A3,64
        0022   0040
00021   0024   5A20               RB(L)     TRY
00022   0026   207F               HLT                                 OBSERVE (
00023   0028   5F2A               RB        START
00024                    *
00025   002A   1501     COUNT     ADK       A5,1                      ** 1.8 **
00026   002C   5F1C               RB        SST                       ** 1.8 **
00027                    *
00028                    ** TT ** TIME IN MICROSECONDS FOR P852M.
00029                    *TOTAL: 5.3 + 1.8 + 1.8 + 1.8 = 10.7
00030                    *
00031                    *ONLY TO GO TO A MONITOR:
00032   002E   2840               ENB
00033   0030   2804               LKM
00034   0032   0003               DATA      3
00035                              END       START
```

This program for the X1215 reads sectors 1--15 from the track where the heads happen to be positioned. It does not read sector 0, to give a better recognizable pattern on the oscilloscope. It was made to check the IOP speed. The number of times the program went through the 10.7 uS loop is recorded in (A5).

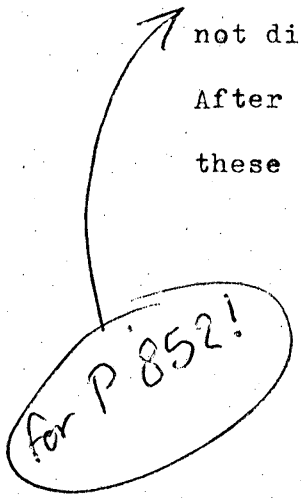
```
PAGE 1     DATE  76-05-21          IDENT     MEMSCN


00000                                IDENT     MEMSCN
00001                          *SCAN THE MEMORY & COMPARE.
00002                          * (A1) MASK
00003                          * (A2) DATA
00004                          * (A3) ADDRESS
00005                          * (A4) WORKING REGISTER
00006                          *
00007   0000   842C           START     LDR*      A4,A3
00008   0002   A404                     ANR       A4,A1
00009   0004   EC08                     CWR       A4,A2
00010   0006   5402                     RF(NE)    *+4
00011   0008   207F                     HLT
00012   000A   1302                     ADK       A3,2
00013   000C   5F0E                     RB        START
00014                          *
00015                          *METHOD: FILL A1 WITH ONES IN THE FIELDS YOU WANT
00016                          *THEN PUT WHAT YOU WANT TO CHECK IN  A2
00017                          *IF EQUALITY IS FOUND A3 CONTAINS THE ADDRESS.
00018                                    END
```

This little program must be keyed in manually. It can be used to find the locations of specific data patterns in the memory.

If you load it starting on address /40 it is most likely that you do not disturb the program(s) that are already in the memory.

After finding one pattern, push start to see if there are more of these patterns.



Data Communication

Some expressions:

-BAUD Number of bits/second. The ASR is a 110 Baud device

-Full Duplex Send and Receive simultaneously, like the normal telephone

-Half Duplex Send or Receive alternating. Used in radio-telephony

-Simplex Only one way, send or receive. Your own radio set!

Apart from the V24 controllers that were discussed in the V24 chapter there are a few more possible with P852-856-857:

ALCU-Asynchronous Medium Speed Line Control Unit for 2 or 4 V24 circuits.

AMA8A-Asynchronous Medium Speed Data Line Multiplexer, 8 units for 50--9600 Baud data lines.

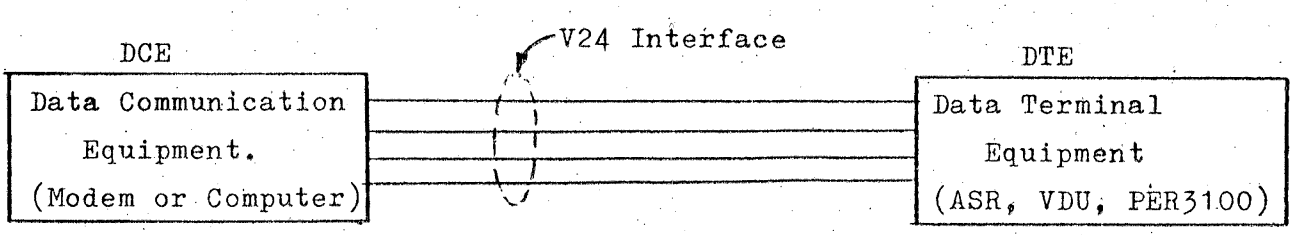
AMA8C-Same, but for TTL or current loop compatible devices

SLCU4-Synchronous line control unit for 4 half duplex or 2 full duplex devices.

AMA16-Low speed C.U. for 16 full duplex lines. Max 200 Baud, TTL compatibl

The codes used depend on the device. In general each code begins with a single start bit. Then come 8 (ASCII) data bits, then either one or two stop bits. The code length is 10 or 11 bits. On 9600 Baud that means almost 900 characters (or bytes) per second.

This is a review of the most important V24 signals



		Signals from DCE to DTE →	P800 Name:
3	104	Received Data	RRD
	119	Received Backward Channel Data	-
5	106	Ready For Sending	CTS
6	107	Data Set Ready	OPE
8	109	Data Channel Received Line Signal Detector	VAL
	110	Signal Quality Detector	-
	112	Data Rate Selector	-
	117	Stand-By Indicator	OPE
	121	Backward Channel Ready	-
	122	Backward Channel Received Line Signal Detector	-
	123	Backward Channel Signal Quality Detector	-
22	125	Calling Indicator	-
15	114	Transmitter Timing	TRE
17	115	Receiver Timing	-
	134	Received Data Present	
		Signals from DTE to DCE ←	
2	103	Transmitted Data	TTD
	118	Transmitted Backward Channel Data	-
4	105	Request to send	RTS
20	108	Connect Data Set to Line	CNS

CCITT Circuit Numbers
connector pins

1	101	Protective ground
7	102	Signal ground

The IEC bus.

The following pages are taken from a description of the PM 4410. This machine is one of the versions of the PM 4400 Compact Computer. The PM 4400 contains a P851 CPU,32K(sometimes 16K) memory, special cards to control a keyboard, a VDU and 1 or 2 mini-floppy discs. This is put together in a special cabinet. The rack inside this cabinet is not the standard P851 rack: the control panel board **can't** be used!

The PM 4410 is the IEC bus controller, it contains the IEC bus card which is part of MIOS 6. (P C 1 7 0 2)

The PM 4420 is the MDS. (Micro processor Development System) It is being designed now (Jan. '79)

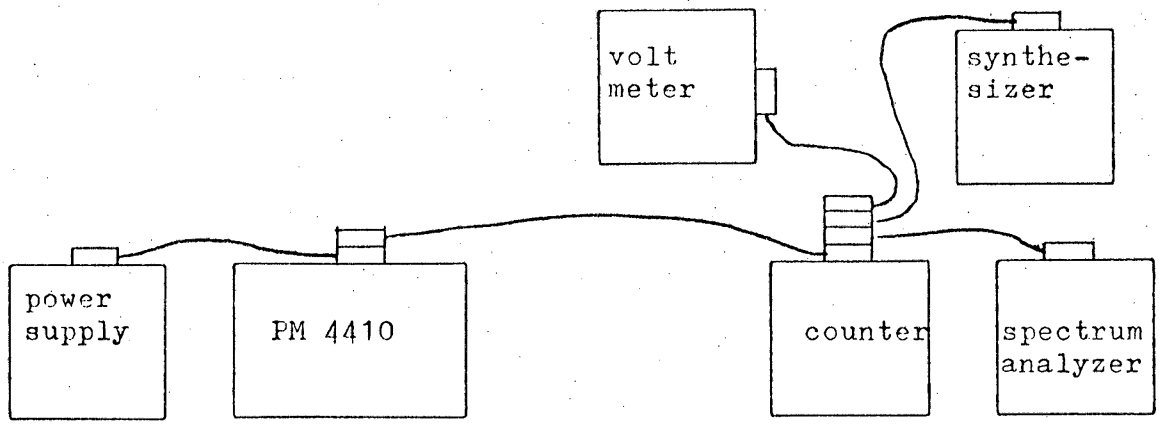
The PM 4430 is the Datacom tester. It is made by Data Systems, Apeldoorn.

Not all information in the following pages is relevant for this Hardware Guide, but it gives an idea of the operation of the IEC bus.

The IEC bus.

The General Purpose Interface Bus (GPIB), IEEE Standard 488-1975 is the same thing as the IEC bus, IEC norm sheet 66 CO 22.
One of its official names is: "Digital Interface for Programmable Instrumentation".
In short it can be said that the IEC norm describes the way to connect maximum 15 programmable instruments with one controller.

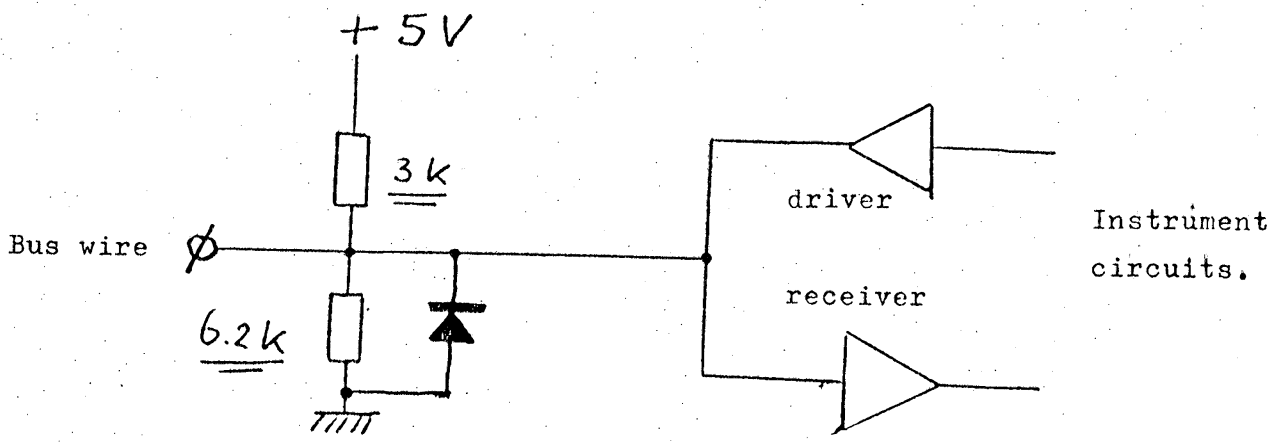
The bus contains 16 signal wires, and a number of ground wires. Usually a 25 pin connector is used; then there will be 9 ground wires. All instruments are equipped with a "male" connector.
The interconnecting cables have a both ends a male and a female connector, so, in total, 4 connectors per cable. This makes it possible to make a daisy chain connection, a star connection, or a combination of both.
No cable may be longer than 4 meters, the total length may not be greater than 20 meters. If many instruments are used, the total length may be 2x(nr. of instruments), if this is less than 20.



Possible configuration.

All lines are:
- active low
- TTL compatible
- open collector (sometimes tri-state can be used)

This means that any line which is in the logical "true" state will have a voltage of 0--0.2 V. If the line is logically "false", it will have a voltage of 2.8--5 V.
Some lines can be driven by different sources. In this case, all sources have to drive the line false, to make it really false.(2.8V)
If any source makes the line true, it will be true.(0.2V)
TTL stands for Transistor-Transistor-Logic, which is the type of circuitry used by most modern digital instruments.

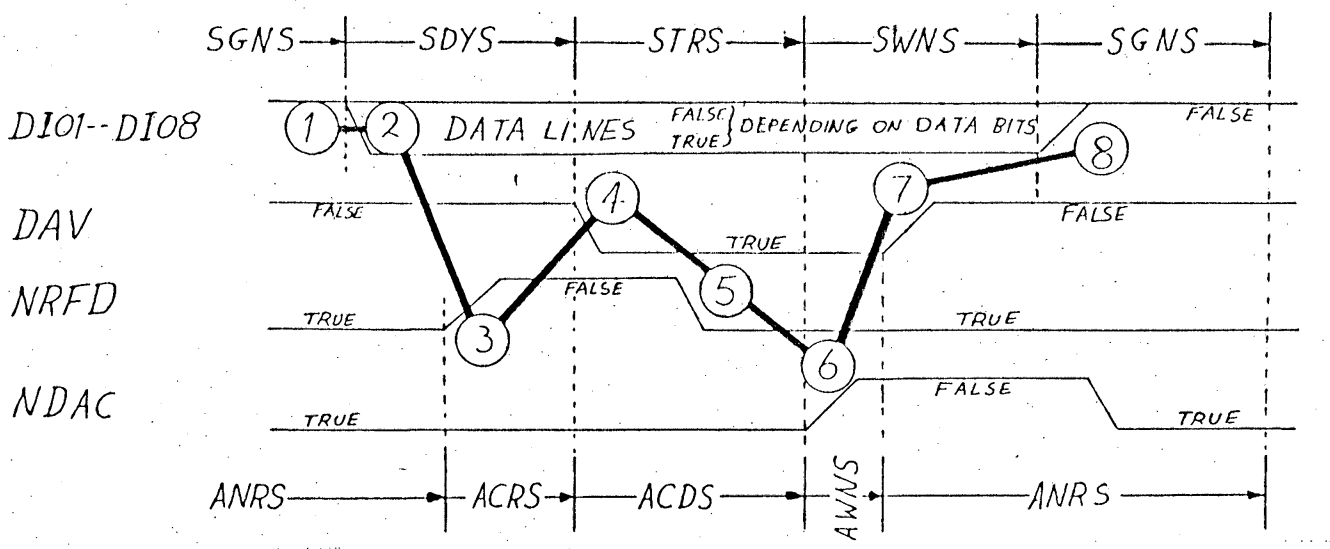


This picture shows a typical circuit to send or receive signals on the bus. Even if in a certain instrument, the driver and/or the receiver is missing, the resistors and the diode will be there. The diode makes sure the bus never goes negative, the resistors determine the impedance and the steady-state voltage of the bus. It will be clear that a device which is turned off will load the bus, as the +5V may be considered to be at ground potential.
For that reason the norm prescribes that at least half of the used instruments plus one must be turned on. If not, marginal operation of the system may result.
The 16 signal wires comprise two groups of 8 wires. The first group contains 8 data lines. These lines are used to send data from and to the instruments and the PM 4410. Also the device number and the command codes can be present on these lines.
The other group of 8 contains control signals.

connector pin number	signal name	direction 4410-inst	explanation	
1	DIO 1	⟷	data, least significant bit.	
2	DIO 2	⟷	⎜	
3	DIO 3	⟷	⎜	
4	DIO 4	⟷	⎜	
14	DIO 5	⟷	⎜	
15	DIO 6	⟷	⎜	
16	DIO 7	⟷	⎜	
17	DIO 8	⟷	data, most significant bit.	
12	ATN	→	Attention	
10	IFC	→	Interface Clear	
5	REN	→	Remote Enable	
6	EOI	←	End, Or Identify	
11	SRQ	←	Service Request	
7	DAV	⟷	Data Valid	} Handshake signals
8	NRFD	⟷	Not Ready For Data	
9	NDAC	⟷	Not Data Accepted	
18--25	Ground			
13	Shield			

IFC is a general clear signal, to reset all instruments on the bus. The effect of REN depends on the device. In some devices the front panel controls are disabled by Remote Enable. When the BASIC statement INIT is performed, both IFC and REN are send to the bus.
The signal ATN is used to alert all devices: the information on the bus must be considered by all devices when ATN is true. Usually this information will be a device address on the DIO lines.
Each device has its own address. This is set up by means of switches or jumper wires in each device. If ATN is true each device compares the state of these wires or switches with the state of the DIO lines. In this way one device may recognize its own address.
When ATN is false the information on the bus is for one device only, this means that addressing should have taken place earlier.
The signal EOI has two uses. At the end of a message of several characters it can be used to indicate the end. It can be used as "identify" during a parallel poll. Parallel poll is not used by the PM 4410 system.

Depending on the direction of the dataflow on the bus, devices are called Listeners or Talkers. Some devices can only listen: a power supply will listen to the setup data. Most devices can both listen and talk: a voltmeter first listens to information about range and sampling time, then, after doing the measurement, it talks to send the measured value on the bus. A third device is the "control element". In this context this will always be the PM 4401 itself. Apart from controlling the bus, the PM 4410 can also operate as a listener or a talker. Operations between devices and the controller are synchronized with the "three line handshake". At any moment there can be only one talker, but there may be several listeners. That is the reason the three line handshake is necessary: first all listeners have to say to the talker that they are ready to hear, then they wait until the talker talks, then they must all say they heard it.



In the timing diagram an example of the handshake is given.
At the top the states of the Source (Talker) are shown, at the bottom those of the Acceptor (Listener).
The timing diagram is only an example, the actual length of the pulses depends on the system components. Also the controller function is not shown.

The digits in the following description refer to the digits in the timing diagram.

1. When the source has data to transfer it goes to the Source Generate State. (SGNS) DAV remains false. The acceptor will be in the Acceptor Not Ready State.(ANRS) The acceptor will make both NRFD and NDAC true. (low)
2. Now the source enters the Source Delay State. (SDYS) The 8 data bits are set on the data lines. Now the source waits until NRFD goes false. (high) Not Ready For Data being false means: Ready for data. There can be several Acceptors at the same time on the IEC bus, if any of them makes NRFD low, the source goes on waiting. This means that the system always waits for the slowest Acceptor.
3. ACRS, the Acceptor Ready State, means that all Acceptors are ready.
4. If NRFD is high, the Source goes in the Source Transfer State. (STRS) DAV is made true. Now the Source waits until the data is accepted.
5. The Acceptor goes in the Accept Data State (ACDS) when it sees DAV true. Because it is then no longer ready to accept more data, it makes NRFD true.
6. When the Acceptor accepts the data, it makes NDAC false. Not Data Accepted being false means: Data is accepted. NDAC will only go false when all Acceptors are making NDAC false. The Acceptor moves to the Wait for New Data State(AWNS).
   When NDAC is false, the Source will go to the Wait for New Data State, too. (SWNS)
7. The Source must make DAV false. This is seen by the Acceptor: it switches to the ANRS again, the initial state.
8. The Source goes to SGNS again, the complete handshake sequence is ready

The device address, as mentioned earlier, is a unique code for one device on the DIO lines with ATN true. The actual address is a 5 bit code, which makes 32 combinations possible. Address 0 and 31 are used for special purposes, the other addresses (1-30) may be used freely. For reasons of maximum bus loading the maximum number of instruments is limited to 15, however.
Two more bits are used to indicate if the address is a listener or a talker address:

. 0 1 x x x x x MLA, My Listen Address
. 1 0 x x x x x MTA, My Talk Address

MLA and MTA are expressions from the IEC norm. If INPUT or PRINT statements are performed for an IEC device, automatically the system software adds one of the two bits to the 5 bit number that is specified in the statement.
Device 0 is the Listener/Talker function of the PM 4410 itself.
Device 31 is used as the Unlisten/Untalk message to all devices, to stop traffic on the bus.

As said before, parallel poll is not used in the PM 4410. The POLL statement causes a serial poll to be performed. That means that all adresses from the list (See POLL statement) are send one-by-one on the DIO lines. When one of the devices is ready for data transfer, it says so in its status byte on the DIO lines. Polling stops then, the status byte is returned to the program as shown in the description of the POLL statement.

One thing which is not described in norm 66 CO 22 is the format of the messages that are used on the bus. At this moment (end '78) norm 66 CO 31 is being prepared. This norm fixes certain types of character strings. It will be used on all new Philips instruments. Other instruments may function in a different way, then it will be necessary to study the device operating manual.
All matters concerning bus timing, handshake, etc., are taken care of by the PM 4410 hard- and software, and of course, the instrument hardware. They will only be a problem when devices not conform the IEC norm are used on the bus.
In this case intensive study of the norm and, if possible, the use of a bus analyzer will be necessary. Both subjects are beyond the scope of this manual.

```
LI

100 INIT
110 PRINT #22,8:
120 INPUT #22:A$
130 PRINT A$

 V-00038E-5

ST  00  0130
```

In this example is shown how a digital voltmeter can be used. A PRINT statement on the IEC bus implies that the device must be a Listener, so the system software adds bit 6. The bit pattern on the bus during addressing will be:

0 0 1 1 0 1 1 0

The secondary address 8 starts a conversion cycle on the voltmeter. A short moment later,(<1 mS) an INPUT statement is done. Now the device should act as a Talker, the address will be:

0 1 0 1 0 1 1 0

Now the PM 4410 acts as a Listener, it makes NRFD false. When the conversion cycle of the voltmeter is finished, it will make DAV true. Now the PM 4410 can do the input. It should be clear that the INPUT statement has to wait for the voltmeter: this could be a waste of computer time. If long calculations have to be done, there are ways to speed up the whole process of measuring and computing; how will be shown later. The string that is delivered by the instrument depends on the type: here the PM 2527 is used. The V is printed when the voltage button has been selected on the meter front panel. The remainder of the string is the voltage and the scale factor

```
LI

100 INIT
110 REM  GET A VALUE FROM THE PM 2527 DIGITAL RMS MULTIMETER
120 PRINT #22,8:
130 INPUT #22:A$
140 PRINT A$
150 LET B$=SUBSTR$(A$,3,6)
160 PRINT B$
170 LET C=STR(B$)
180 PRINT C
190 LET D$=SUBSTR$(A$,10,2)
200 PRINT D$
210 LET E=STR(D$)
220 PRINT E
230 LET V=C*10**E
240 PRINT V
```

```
 V+02856E-4
+02856
    2856
-4
       -4
         .2856

ST  00  0240

 V-12204E-5
-12204
   -12204
-5
       -5
        -.12204

ST  00  0240
```

This is the result of the previous program. If the values from the meter are only used for logging purposes, the string as received will do. In all other cases an arithmetic variable is easier to handle.
Most of the printing in the example supplies redundant information. It is only there to show what each step of the program does.

```
100 INIT
110 LET VT=0
120 FOR I=1 TO 10
130 PRINT #22,8:
140 INPUT #22:A$
150 LET C=STR(SUBSTR$(A$,3,6))
160 LET E=STR(SUBSTR$(A$,10,2))
170 LET V=C*10**E
180 LET VT=VT+V
190 NEXT I
200 PRINT TIME$;"   AVERAGE: ";VT/10
210 GOTO 100
```

In this program the data conversion in done completely in the lines 150, 160 and 170. Line 180 is used to compute the sum of 10 readings. This is used for printing the average only. Because the time is also printed, some idea about the speed can be obtained. The shortest time between printing is about 3 S, the longest time is 6 S. This is caused by the automatic range selection of the voltmeter. If it has to change its range, the conversion time is longer. The average time is (51-18)/(10x8)=0.4 S

```
095318   AVERAGE:      -.115517
095324   AVERAGE:       .043454
095327   AVERAGE:       .50623
095331   AVERAGE:       .63202
095334   AVERAGE:       .696
095338   AVERAGE:       .72495
095341   AVERAGE:       .81626
095345   AVERAGE:       .83864
095351   AVERAGE:       .157705
```

The actual output of the program went on the display unit, so hardly any time was lost. The hard-copy on this page was obtained by pushing the copy button.

Integrated Circuits

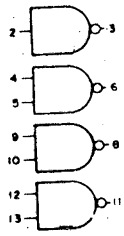
I tried to make a list of all I.C.'s in the P800 computers and control units.

		IC Guide, page nr:
1488	LINE DRIVER, QUAD	1
1489A	LINE RECEIVER, QUAD	1
1801	SEE: 7400, OPEN COLLECTOR	1
2502	SEE: UART	8
2601	SEE: USR/T	
2681	SEE: 74188	3
2691	SEE: 74188	3
2701	SEE: 74188	3
2711	SEE: 74188	3
2721	SEE: 82129	6
2841	FIFO STACK, 64X4 BIT	1
3101	MEMORY, 16X4 BIT	1
5016	BAUD RATE GENERATOR	1
6200	1024 BIT ROM(256X4)	1
7400	NAND, QUAD 2 INPUT	1
7402	NOR, QUAD 2 INPUT	1
7403	SEE: 7400; OPEN COLLECTOR	1
7404	INVERTER, HEX	1
7405	SEE: 7404, OPEN COLLECTOR	1
7406	SEE: 7404, 30 V. OPEN COLLECTOR	1
7407	AMPLIFIER, HEX	1
7408	AND, QUAD 2 INPUT	1
7409	AS 7408, OPEN COLLECTOR	1
7410	NAND, TRIPLE 3 INPUT	2
7411	AND, TRIPLE 3 INPUT	2
74112	J-K FLIP-FLOP, DUAL	2
74121	ONE-SHOT MULTIVIBRATOR(9603)	2
74125	BUS DRIVER, QUAD 3-STATE	2
74132	NAND, QUAD 2 SCHMITT TR. INP.	2
74133	NAND, SINGLE 13 INPUT	2
74136	X-OR, QUAD OPEN COLLECTOR	2
74138	DECODER, OCTAL	2
7414	SEE: 7404	1
74148	PRIORITY ENCODER(9318)	2
74151A	MULTIPLEXER SINGLE 8 INPUTS	2
74153	MULTIPLEXER, DUAL 4 INPUT	2
74157	MULTIPLEXER, QUAD 2 INPUTS	3
74158	MULTIPLEXER, QUAD 2 INPUTS, IN	3
7416	SEE: 7404, 15 V. OPEN COLLECTOR	1
74161	SYNCHRONOUS COUNTER, 4 BIT	3
74164	SHIFT REGISTER, 8 OUTPUTS	3
74169	COUNTER, 4 BIT UP/DOWN	3
7417	AMPL, HEX BUFFER/DRIVER OPEN COLL	3
74170	SCRATCHPAD, 4X4 BIT	3
74174	D FLIP-FLOP, 6X	3
74175	D FLIP-FLOP, 4X	3
74181	ALU, 4 BIT (9341)	3
74182	LOOK-AHEAD CARRY GENERATOR	3
74188	PROM, 32X8 BIT	3

74194	SHIFT REGISTER, 4 BIT	4
74195	SEE: 9300	7
74197	HEXA COUNTER, PRESETTABLE	4
7420	NAND, DUAL 4 INPUT	4
7425	NOR, DUAL 4 INPUT STROBED	4
7426	SEE:7400, HIGH VOLTAGE	1
7427	NOR, TRIPLE 3 INPUT	4
74279	LATCH, QUAD S-R	4
7428	NOR, QUAD 2 INPUT	4
74281	ACCUMULATOR, 4 BIT	4
74298	MULTIPLEX REGISTER, QUAD 2 INP	4
7430	NAND, SINGLE 8 INPUT	4
7432	OR, QUAD 2 INPUT	4
7433	SEE: 7428, OPEN COLLECTOR	4
7437	SEE: 7400; BUFFER	1
7438	SEE: 7400	1
7440	SEE: 7420	4
7450	AND-OR-NOT, EXP. DUAL 2X2	5
7451	AS 7450; NO EXPANDER	5
7453	AND-OR-NOT, EXP. SINGLE 4X2	5
7454	AS 7453; NO EXPANDER	5
7464	SEE: 74S64	5
7474	D FLIP-FLOP, DUAL	5
7475	LATCH, QUAD	5
7476	J-K FLIP-FLOP, DUAL	5
7485	COMPARATOR, 4 BITS	5
7486	X-OR, QUAD 2 INPUT	5
7488A	256 BIT ROM(32X8)	5
7489	64 BIT RAM(16X4) (93403)	5
7493	BINARY COUNTER, 4 BIT	5
74S00	SEE: 7400	1
74S04	SEE: 7404	1
74S11	SEE: 7411	2
74S112	SEE: 74112	2
74S138	SEE: 74138	2
74S169	SEE: 74169	3
74S174	SEE: 74174	3
74S175	SEE: 74175	3
74S20	SEE: 7420	4
74S287	PROM 1024 BIT	5
74S30	SEE: 7430	4
74S64	AND-OR-NOT, 4-3-2-2 INPUTS	5
74S74	SEE: 7474	5
75365	TTL-->MOS DRIVER	6
75451	PERIPHERAL DRIVER, DUAL	6
7812C	VOLTAGE REGULATOR, +12V.	6
7815C	VOLTAGE REGULATOR, +15V.	6
7905	VOLTAGE REGULATOR, -5V.	6
7912	SEE: 7812	6
8205	ROM, 4096 BIT (512X8)	6
82129	ROM 1024 BIT	6
8234	MULTIPLEXER DUAL 4 INPUT, OPEN COLL	6
82S123	ROM, SPECIAL P851	6
8576	PROGRAMMABLE LOGIC ARRAY	6
8838	BUS RECEIVER/DRIVER	6
8T28B	BUS RECEIVER/DRIVER 3-STATE	6

8T97	BUS DRIVER, 3-STATE	7
9300	SHIFT REGISTER 4 BIT	7
9301	ONE OF TEN DECODER	7
9308	LATCH, DUAL 4 BIT	7
9309	MULTIPLEXER, DUAL 4 INPUT	7
9310	SEE: 74161	3
9316	SEE: 9310	3
9318	SEE: 74148	2
9324	COMPARATOR, 5 BIT	7
93403	SEE: 7489	5
9341	SEE: 74181	3
9602	ONE SHOT, DUAL	7
9603	SEE: 74121	2
ADL2011	SEE: 74188	3
AM9101BPC	STATIC RAM, 256X4 BIT	7
BMTT	TRANSLATOR, CMOS<---->TTL	7
CR202A	SEE: 74188	3
LDM100	DIGITAL DELAY MODULE	7
LDM250	DIGITAL DELAY MODULE	7
NE555V	TIMER	7
OQ003	SCHMITT TRIGGER, HEX	8
OQ6682	PLANET, SPECIAL P851	8
OQ6683	SPALU, SPECIAL P851	8
REC0612	SEE: 7407	1
REC0613	SEE: 7404	1
TERNET	BUS TERMINATOR	8
TIL111	OPTO-COUPLER	8
UA710DC	DIFFERENTIAL COMPARATOR	8
UA723	VOLTAGE REGULATOR	8
UART	UNIVERSAL ASYNCHRONOUS RECEIVE TR	8

1488



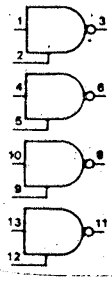
1489A

LINE RECEIVER

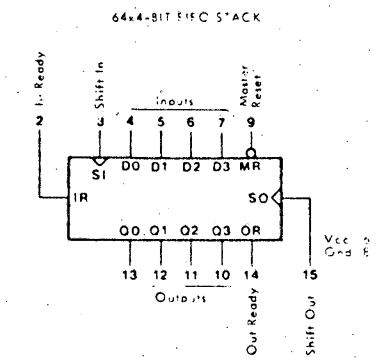
Response control gives:

a) Logic threshold shifting

b) Input noise filtering

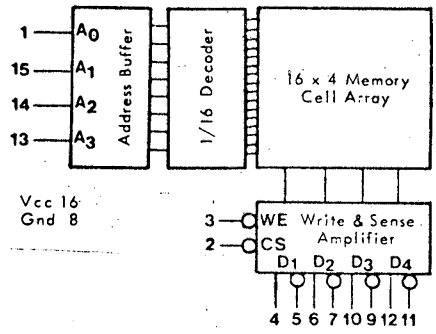


2841

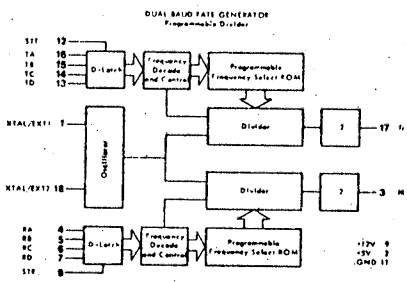


3101

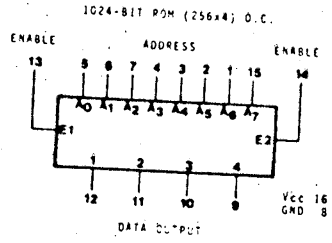
HIGH SPEED FULLY DECODED 64 BIT MEMORY



5016



6200

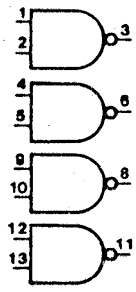


7400

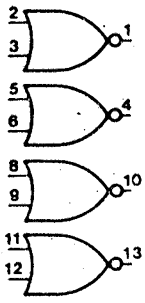
7400
S00
H00
03[1]
37[2]
38
1801[1]

1 open collector

2 buffer

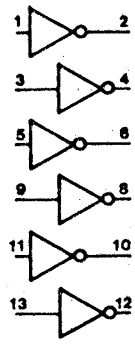


7402

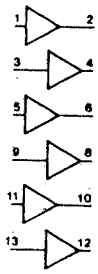


7404

7404, S04, H04
7406 o.c. 30V
7414 sh Tris
7416 o.c. 15V
7405 o.c. 5V



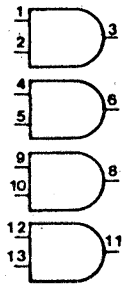
7407



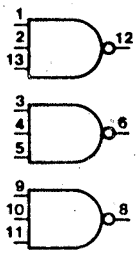
7408

7408
09[1]

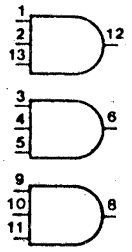
1 open collector



7410



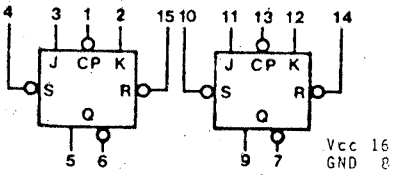
7411



74112

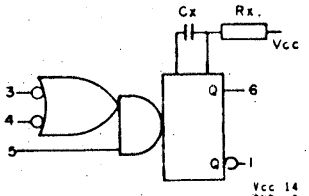
74S112

J-K NEGATIVE-EDGE-TRIGGERED FLIP-FLOPS



74121

SINGLE SHOT WITH SCHMITT TRIGGER INPUT

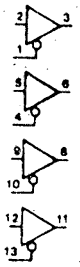


FUNCTION TABLE						
t_n INPUT		t_{n+1} INPUT			OUTPUT	
L	X	L	L	X	H	Trigger
X	L	L	X	L	H	Trigger
H	H	H	X	L	H	Trigger
H	H	H	L	X	H	Trigger

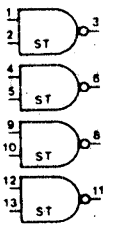
* Triggering Transition

74125

BUS BUFFER WITH 3-STATE OUTPUT



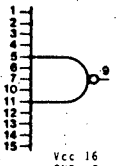
74132



Vcc 14
GND 7
Positive going threshold V+1.7
Negative going threshold V+0.9

74133

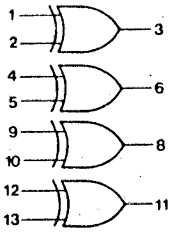
13-INPUT NAND GATES



Vcc 16
GND 8

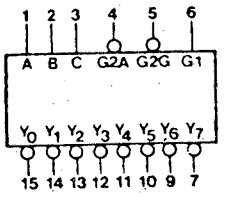
74136

QUADRUPLE 2-INPUT EXCLUSIVE-OR GATE WITH OPEN-COLLECTOR OUTPUTS



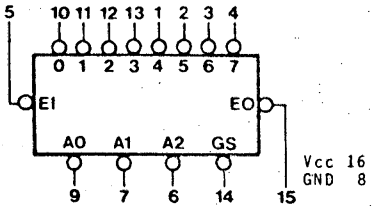
74138

DECODER



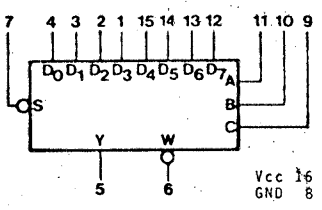
74148

8-LINE-TO-3-LINE PRIORITY ENCODER

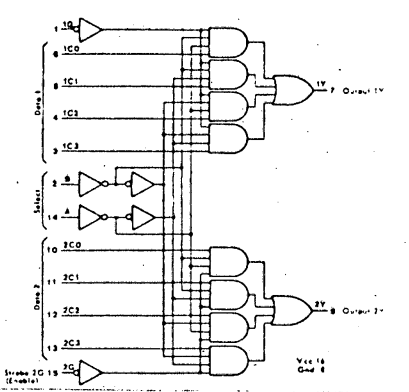


74151A

DATA SELECTOR/MULTIPLEXER

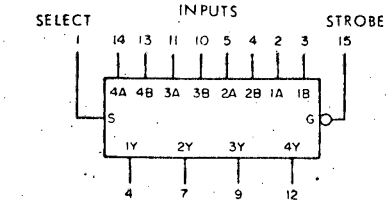


74153

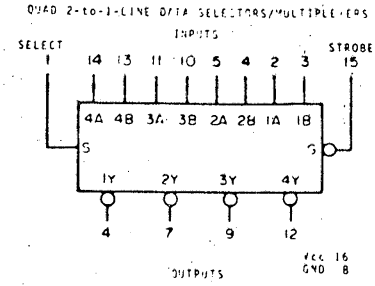


74157

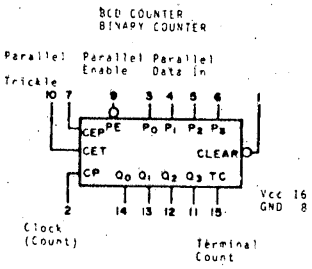
QUAD 2-to-1-LINE DATA SELECTORS/MULTIPLEXERS



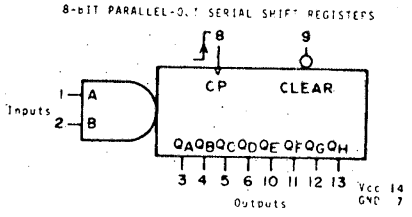
74158



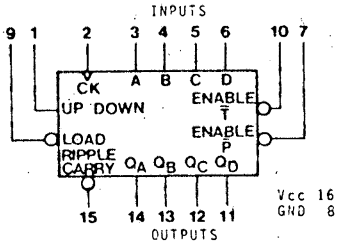
74161



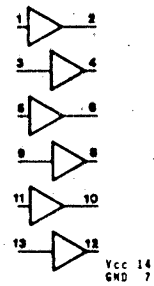
74164



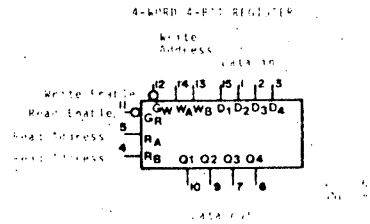
74169



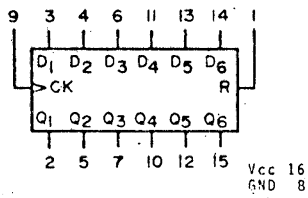
7417



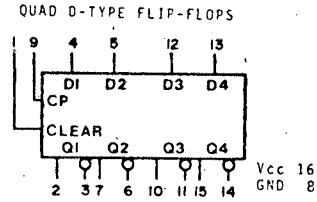
74170



74174

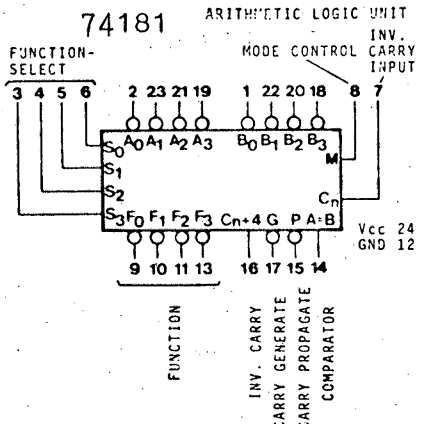


74175



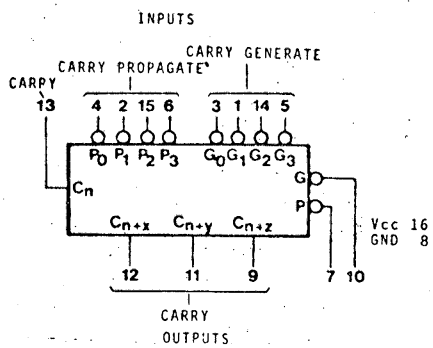
74181

ARITHMETIC LOGIC UNIT



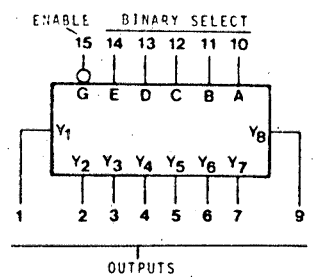
74182

LOOK-AHEAD CARRY GENERATOR



74188

256-BIT PROGRAMMABLE READ-ONLY MEMORY



Organized as 32 Words of 8 Bits Each

74194

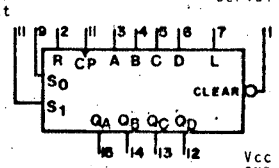
4-BIT SHIFT REGISTER

Shift Right Serial Input

Parallel Inputs

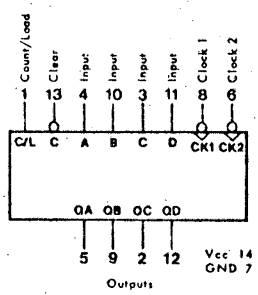
Shift Left Serial Input

Mode Select



74197

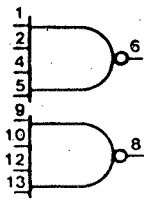
50/30 MHz PRESETTABLE DECADE OR BINARY COUNTER/LATCH



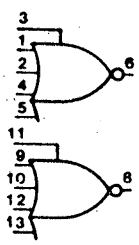
7420

7420 7440

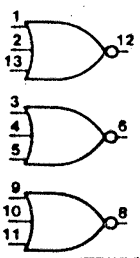
BUFFER



7425

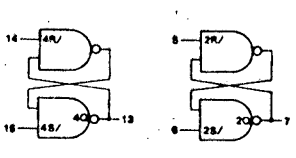


7427



74279

QUADRUPLE S-R LATCHES



FUNCTION TABLE

INPUTS S*	R	OUTPUT Q
H	H	Q0
L	H	H
H	L	L
L	L	H*

Vcc 16
GND 8

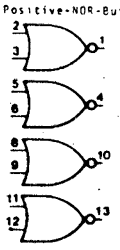
* For latches with double S inputs:
H = both S inputs high
L = one or both S inputs low

Q0 = the level of Q before the indicated input conditions were established.

* This output level is pseudo stable; that is, it may not persist when the S and R inputs return to their inactive (high) level.

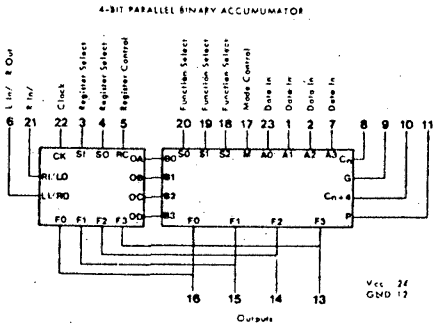
7428

2-Input Positive-NOR-Buffer



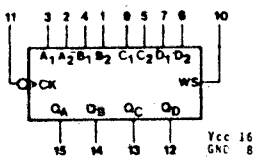
74281

4-BIT PARALLEL BINARY ACCUMULATOR



74298

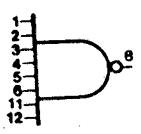
QUADRUPLE 2-INPUT MULTIPLEXER WITH STORAGE



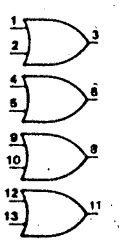
FUNCTION TABLE

INPUTS		OUTPUTS
WORD SELECT	CLOCK	QA QB QC QD
x	↑	a1 b1 c1 d1
w	↑	a2 b2 c2 d2
y	↓	QA0 QB0 QC0 QD0

7430



7432

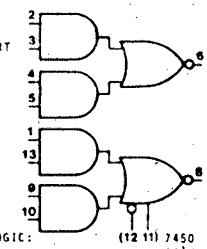


7450

7450 (EXPANDABLE)

7451

2-WIDE
2-INPUT
AND-OR-INVERT



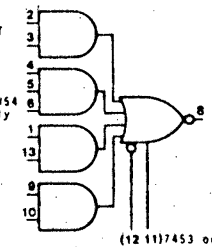
POSITIVE LOGIC: (12 11) 7450 only

7453

7453 (EXPANDABLE)

7454

4-WIDE
AND-OR-INVERT

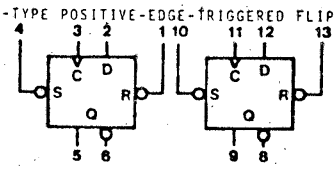


(12 11)7453 only

7474

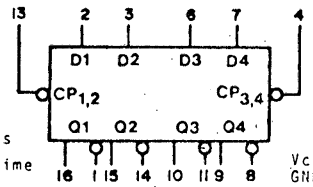
7474, H74, S74

DUAL D-TYPE POSITIVE-EDGE-TRIGGERED FLIP-FLOP



7475

4-BIT LATCH

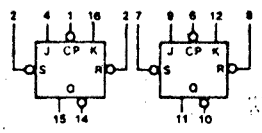


CP is Store Time

Vcc 5
GND 12

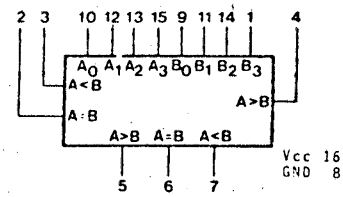
Q follows D when CP is high

7476



7485

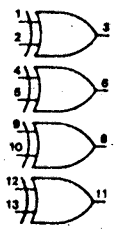
4-BIT MAGNITUDE COMPARATOR



Vcc 16
GND 8

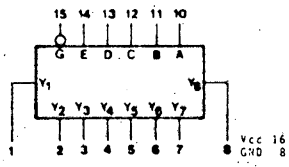
7486

QUADRUPLE 2-INPUT EXCLUSIVE-OR GATES



7488A

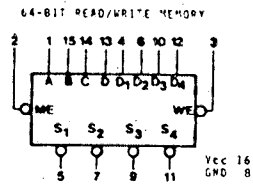
256-BIT READ-ONLY MEMORIES



Vcc 16
GND 8

7489

64-BIT READ/WRITE MEMORY

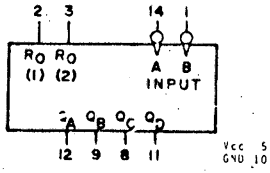


Vcc 16
GND 8

ME	WE	OPERATION	CONDITION OF OUTPUTS
L	L	Write	Complement of Data Inputs
L	H	Read	Complement of Selected Word
H	-	Inhibit Storage	Complement of Data Inputs
H	H	Do Nothing	High

7493

4-BIT BINARY COUNTER



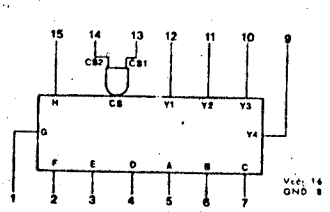
Vcc 5
GND 10

$R_0(1)$, $R_0(2)$ = Reset to zero

Input A is input to flip-flop A
Input B is input to 3-bit binary counter (flip-flops B, C, and D). For 4-bit counter operation input B is connected to output QA.

74S287

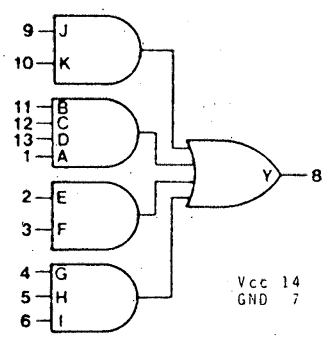
1024 BIT PROGRAMMABLE READ-ONLY MEMORY



Vcc 14
GND 8

74S64

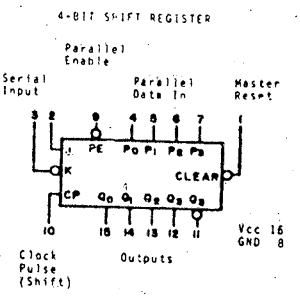
4-2-3-2-INPUT AND-OR-INVERT GATES



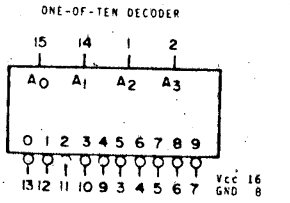
Vcc 14
GND 7

8T97

9300



9301

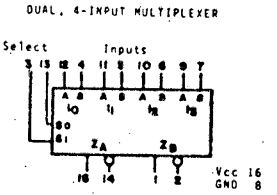


Multipurpose Decoder will accept four inputs and provide 10 mutually exclusive outputs.

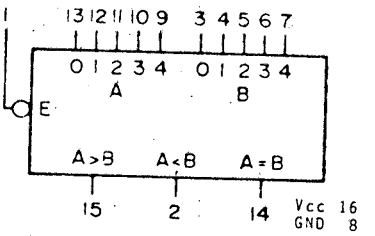
9308

load dual

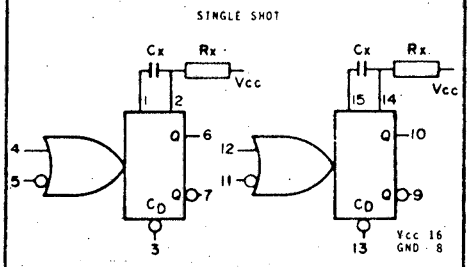
9309



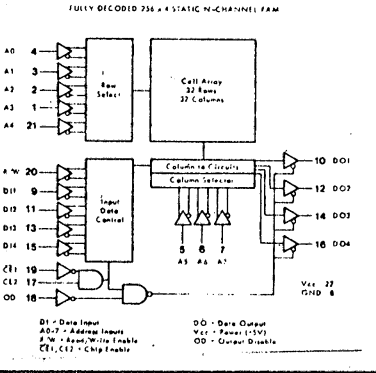
9324



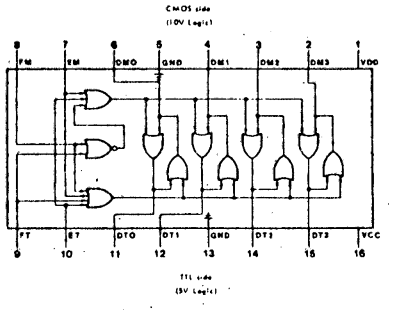
9602



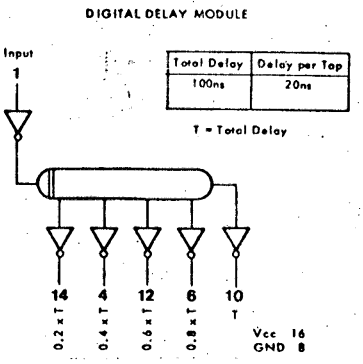
AM9101BPC



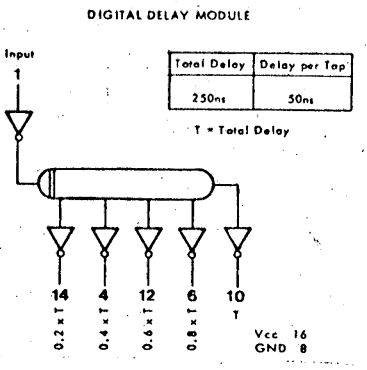
BMTT



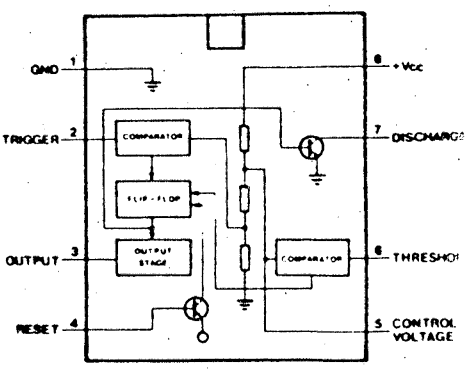
LDM100



LDM250



NE555V



0Q003

6x Schmitt trigger high voltage

0Q6682

Special P8S1

0Q6683

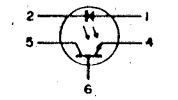
Special P8S1

TERNET

Bus terminator

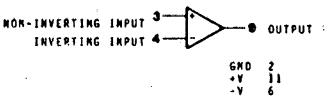
TIL111

OPTICALLY COUPLED ISOLATOR



UA710DC

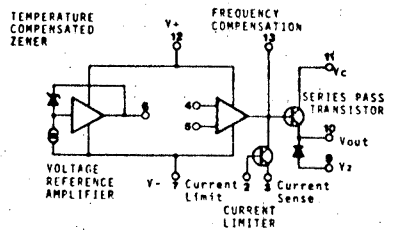
HIGH SPEED DIFFERENTIAL COMPARATOR



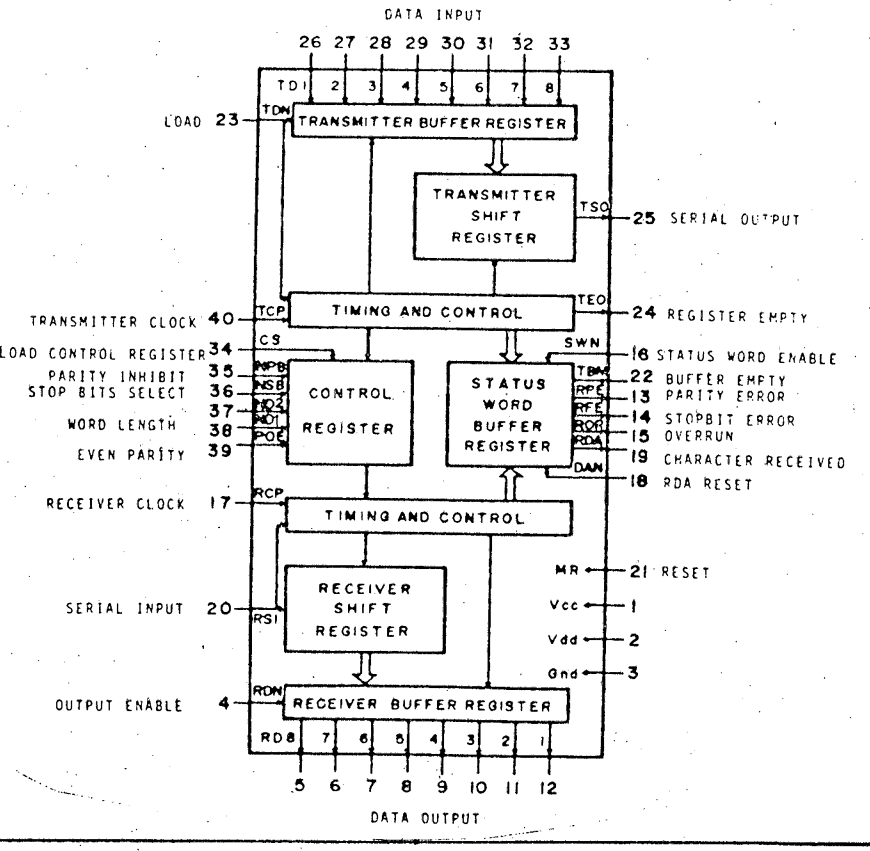
UA723

μA723

PRECISION VOLTAGE REGULATOR

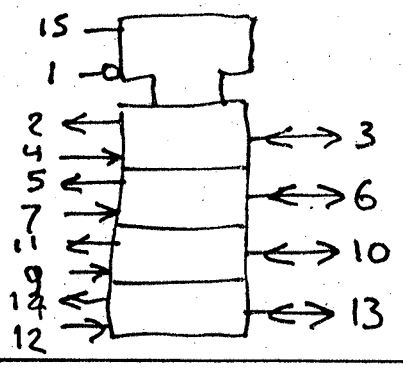


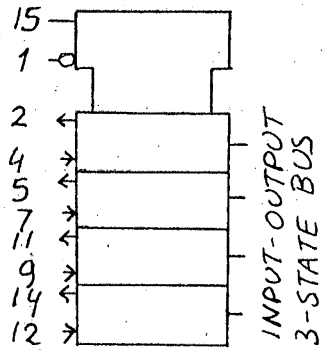
UART



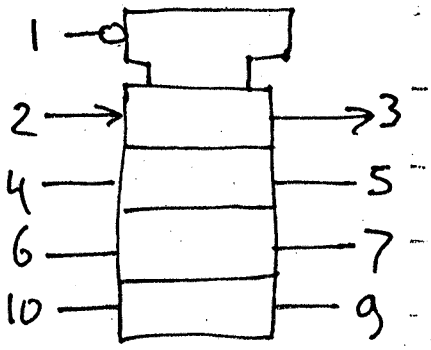
IC Guide, 9. (preliminary)

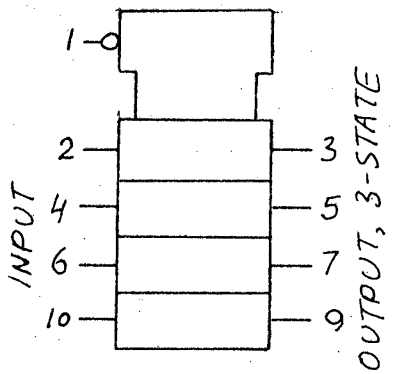
8T28B



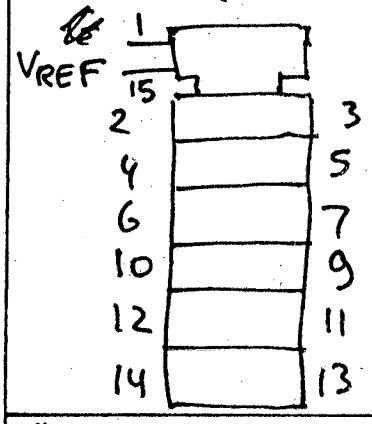


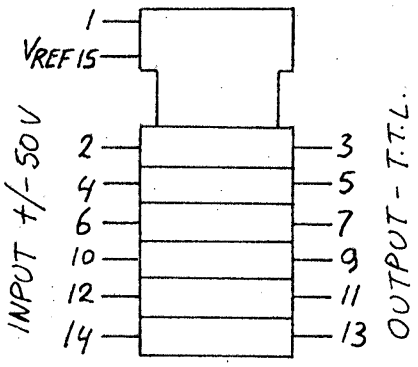
8T97





OQ003





New P800 hardware in 1980.

At this moment (Jan '80) P851, P856 and P857 are standard machines.
To be introduced in 1980 are:

P858 board. This board contains a CPU (performance 1.5x P857), one V24, one MMU and 2 IOP channels

P859 system. This system contains a board similar to the P858, but all memory is MOS, max. 128K words per board. The addressing capability is extended to 512K words (4 memory boards). So, with say 8 boards, a very powerful system can be built.

The following products use double Euro boards, and the FAST 16 bit microprocessor (P851 compatible)

P853 Similar to P851, but cheaper

P854 Adresses up to 512K words. On the board is a 16 subchannel IOP.

All machines use standard P800 software. P859 and P854 use an upgraded MAS, because of their larger addressing capabilities.

Finally there will be a Single Board Computer, using FAST.
On one board the SBC has:

CPU
V24
4 K RAM
6 K EPROM (12 K in future)
Timers
Parallel I.O. ports

Several new control units will be available on Euro boards. One very important c.u. is the X1215/X1216 disc controller.
Another interesting c.u. is the 4x SCUZ board.