EINDHOVEN UNIVERSITY OF TECHNOLOGY

Design and implementation of a central instruction processor with a multimaster bus interface

Citation for published version (APA):
Fabian, K. (1981). *Design and implementation of a central instruction processor with a multimaster bus interface*. (EUT report. E, Fac. of Electrical Engineering; Vol. 81-E-117). Technische Hogeschool Eindhoven.

Document status and date:
Published: 01/01/1981

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Download date: 10. Jun. 2024

th e

[illegible] University of Technology
Nottingham

Department of Electrical Engineering

Design and implementation of a central instruction processor with a multimaster bus interface

by

K. Fabian

E I N D H O V E N U N I V E R S I T Y O F T E C H N O L O G Y

Department of Electrical Engineering

Eindhoven The Netherlands

DESIGN AND IMPLEMENTATION OF A CENTRAL INSTRUCTION PROCESSOR WITH A MULTIMASTER BUS INTERFACE

By

K. Fabian

TH-Report 81-E-117

ISBN 90-6144-117-X

Eindhoven

February 1981

Abstract

Functional description and a circuit analysis of a Central Instruction Processor based on a CPU Signetics SPC 16/10 is provided. Multimaster bus interface enables the use of the microcomputer in the multiprocessor applications based on the UPL - Bus structure.

Fabian, K.
DESIGN AND IMPLEMENTATION OF A CENTRAL INSTRUCTION PROCESSOR WITH A MULTIMASTER BUS INTERFACE.
Department of Electrical Engineering, Eindhoven University of Technology, 1981.
TH-Report 81-E-117

Adress of the author

Ing. Karol Fabian, CSc
Institute of Cybernetics
Slovak Academy of Science
Horna Strieborna 8
BANSKA BYSTRICA
974 00
Czechoslovakia

Acknowledgement

This work would not be possible without the help of Ir.M.P.J. Stevens and Ing. C.H. Hooidonk and supervision of Prof.Ir. A. Heetman.

Contents

		Page
1.	Introduction	1
2.	Central Processing Unit - SPC 16/10	2
2.1.	General Information	2
2.2.	Memory Organization	3
2.3.	Automatic Restart and Power Fail Interrupt	4
2.4.	I/O System	6
2.5.	CPU Timing	6
2.6.	Instruction Set	6
2.7.	CPU Support Circuits	7
3.	Clock Circuits	8
4.	Input/Ouput	8
4.1.	Serial I/O	8
4.2.	Parallel I/O	9
5.	Interrupt Control	10
6.	Memory Organization	12
7.	Bus Interface	13
8.	DMA Control	15
	Literature	16
	Figures	17

1.Introduction.

This report contains a functional description and a circuit analysis of a single board computer, based on the Signetics SPC 16/10 CPU chip. The microcomputer was designed, built and measured during a UNESCO fellowship which took place from october 1980 till february 1981 at Technical University Eindhoven.
Figure 1 is a simplified block diagram of the microcomputer, that ilustrates the functional interface between the SPC 16/10 CPU and the system facilities via the UPL - Bus (Unified Product Line Bus proposed by Signetics).

According to Signetics convention, low active signals at Signetics chips are described with a letter N to make the report readable together with the used literature. Signals and lines which continue on other figures are described by name and number of figure where they should continue.

2. Central Processing Unit - SPC 16/10.

2.1. General information.

Signetics SPC 16/10 microprocessor is a 16 bit single chip CPU, which can work with SPC 16/12 Unified Bus Manager (UBM) to support multimaster bus systems that share acces to common system resources, and together with the SPC 16/11 Interrupt Handling Controller (IHC) provides a interrupt structure, which supports up to 64 hardware and software interrupt levels. The architecture of the SPC 16/10 is based on an array of sixteen 16-bit registers, one of which is reserved as the program counter, and a second as the system start pointer. The other 14 registers are general prupose and can be used as data accumulators, index registers, or memory pointers.

A 16-bit program status word provides status and control information such as current interrupt level, interrupt control and condition and can be operated upon by certain instructions to alter the operational state. The CPU can directly address 32K 16-bit words software memory and 32K 16-bit words firmware memory, 256 external registers and 64 I/O devices. The firmware memory can be used for functions such as self-check routines and control panel programs. These resourecs decrease system program and read/write memory needs, and can increase the ststem throughput. The instruction set features eight addressing modes and operates on bit, characters (byte), word and double word data types. The instruction set includes hardware multiply and divide, multiple shifts and rotate, load and store of multiple registers and test and set/reset bit operations on bit strings.

The addressing modes include direct, register, indirect, indexed and indexed indirect types. Operations directly on memory words are also supported. For use by the progrmmer are sixteen registers A0 through A15 available. Register P (A0) is used as a program counter. It is incremented in steps of two if the program is to continue in sequence or altered to hold the new address if a branch is to be performed. Working register A1-A14 may be used in the following ways:

as the source of one or both operands required by an instruction.

as a destination of the result of the execution of an instructon.

as a pointer where the specified register contains the operand address rather than the operand itself.

as index registers.

as user stack pointers, which are automatically updated on linking to and return from subroutines.

A15 is the system stack pointer. A15 can be adressed by instruction in the same way as registers A1 through A14 but only while operating in system mode.

PSW register is divided into three parts which together form one 16-bit word (Program Status Word). Certain instruction and hardware actions cause this word to be stored in a memory stack whenever it is required to be saved. Program action is then necessary to restore the saved word. The content of the PSW can be altered by processor intructions while operating in system mode.

PSW register format

PL Priority level						CR		GF		Machine status bits					
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
P0	P1	P2	P3	P4	P5	C0	C1	RUN	ENB	Not used				F	U

PL register - The 6 bit program level register indicates the priority level of the running program.

CR register - The 2 bit condition register holds the state of the result of, or the response to, certain instructions.

GF register - The general flag register contains 8 bit, 4 unused. Remaining bits indicate and control the CPU status.

RUN - The CPU is in the idle state (RUN=0). RUN=1 indicates that the CPU is executing instructions.

ENB - Interrupts are enabled when ENB=1.

F - The CPU executes instructions from firmware memory when F=0.

U - System state is indicated by U=0 and user state by U=1.

2.2 Memory Organization.

The SPC 16/10 addresses two program and data memory spaces referred to as software memory and firmware memory. The

memory currently addressed is indicated internally by the state of the F-bit in the PSW register and externally by the state of the FIRMN output pin. User application programs normally reside in software memory while firmware memory can be used for storage of system routines such as initial program load, automatic restart, operator's console and diagnostics. However it is not necessary to use two distinct memories, if the FIRMN output is not used in address decoding, all mamory accesses will be from a single memory regardless of the state of the F-bit.

Firmware state is entered on receipt of the automatic restart or EPIN interrupts. While in firmware state, the EPIN Interrupts are disabled. All instructions are available in this state and the extended load ELR, extended store ESR and return from firmware RTF instruction are available in the firmware state only. The software state is entered from the firmware state upon execution of an RTF instruction with appropriate parameters. Each address space is 32K words of 16 bits.

Although the CPU data types include bits and characters (bytes) as well as words, accesses to the memory are always in words (even address) and therefore the least significant bit of the address is not revelant for memory addressing. Hexadecimal addresses 0 and 2 of the firmware memory are reserved for automatic restart and EPIN interrupt vectors, while the high addressed (FFFC and FFFE if a full 32K words are in use) are reserved as a save area for the PSW register and program counter contents when EPIN interrupt is received. In software memory, 0000H through 007EH contain address vectors for interrupts and traps. The system stack begins at the initial value of A15 and fills the value if the stack pointer goes lower than 0100H.

2.3 Automatic Restart and Power Fail Interrupt.

Active low input on reset pin sets the CPU to a known internal state. The PWFN input must be low while RESETN is low in order to properly reset the chip. When RESETN gors high after being low for the specified time period, the device executes an internal initialization routine which resets all interrupt flip-flops and initializes PSW register as follows: PLR = 111111, RUN = 0, ENB =1, U = 0 and F = 1. The CPU then remains in an idle condition and does not execute program instruction until PWFN goes high or EPIN goes low.

The negative edge of PWFN input causes a power fail interrupt to be grnerated (level 0). The CPU completes execution of the current instruction, stores P and PSW in the

system stack and writes a new program level (0) into the PL-register and to external register FFH. It then vectors to the first instruction of the interrupt routine, whose address is contained in the word 0 of the memory.
The positive edge of PWFN minimum time t = 20ms after the positive edge of RESETN causes an automatic restart interrupt (level 0) to be generated. If ENB = 1 or subsequently when interrupts are enabled, the CPU switches to firmware mode and vectors to the address contained in word 0 of the firmware memory.
The CPU has the capability of dealing with 62 interrupt levels and two traps. The sources of these interrupts are:

Level	Type of interrupt
0	power failure/automatic restart
1	LKM instruction or system stack overflow
2	INT2N input
3	INT3N input
4 to 61	INTXN input

Associated with each interrupt is a priority ranging between 0 and 61. Since the INTXN input encompasses interrupt levels 4 to 61 external circuitry compare the priority of the interrupting device with the current program level, before activating INTXN. This is acomplished via the SPC 16/11 Interrupt Handling Controller, which function will be described in next chapter. Traps are high priority interrupts that differ from normal interrupts in that they are recognised regardless of the state of ENB bit and independent of program level. The two traps are: Trap 62 is activated upon detection of an instruction with content 68XXH. Trap 63 occurs on other illegal instructions or detction of a privileged instruction while operating in user mode. Location 0000H to 007EH of software memory contains a table of 63 addresses. When an interupt is recognised, the CPU suspends execution of the current program, saves necessary parameters in the system stack and perform an indirect branch via the appropriate table entry to the handling routine.

2.4 I/O System.

The SPC 16/10 can use three methods to transfer data to I/O devices. The first technique, memory-mapped I/O requires, that a portion of the memory space is assigned to I/O devices. Special I/O instructions are also available to address up to 256 external registers and 64 control units, independent of the memory. These two types differ in that the control unit instruction set the condition register to reflect the status of CACK and AREN input pins wile the external registers instruction do not. External register FFH is reserved for use by the interrupt system.

2.5 CPU Timing.

Lit. [1] provides figures to illustrate basic read and write cycles of the CPU. The CLK input provides the basic timing signal. The cycle begins with a generation of the address strobe ASTRO, the trailing edge of which samples the multiplexed address on the AD0-AD15 lines. This address together with the state of the AMA, AMB and WRITE outputs determines the device being accessed and whether a read or write operation is being performed. In the next clock cycle REQN output goes low to alert the external devices that a data transfer operation is in progress. At the same time, data is placed on the AD outputs for a write operation, or the lines are prepared for input for a read operation.

One and a half clock periods later the state of the RDY input line is sampled to determine if the addressed device has accepted or provided the data. If RDY is low, a wait cycle is inserted and the process is repeated one clock period later. When RDY goes high the cycle is terminated. Data and status information is accepted by the CPU if required, and REQN is returned to high, to indicate that the transfer has been completed.

2.6. Instruction set.

SPC 16/10 executes a powerful set of instructions, which is described in [7]. It provides eight addressing modes, bit, character (byte), single and double precision integer and logical data types.

2.7. CPU suport circuits.

CPU support circuitry consists of the cycle status decoder, which is based on Intel 8205 chip. The inputs to this decoder are AMA, AMB, REQN and WRITE signals produced by the CPU unit. Output control signals are connected to DM 8097 3 - state buffer. This buffering allows disconection of these control signals from the control bus by the DMA cycle. 3-state of the DM 8097 is entered with the AEN signal from the DMA controller. Low active control signals provided by this scheme include MEMRD, MEMWR, I/OR, I/OW, ERR, ERW. Two types of MEMRD and MEMWR control signals are provided. One for normal memory read/write cycle and second pair for memory access with lock. This is useful in multimaster systems for implementations of semaphores. The memory address with lock specification is generated during tne execution of TSB (test bit) and TRB (reset bit) instructions.

Address and data are demultiplexed by the means of a pair of Intel 8282 latches and a pair of 8286 buffers. 8282 chips are used for address latching after the trailing edge of ASTRO strobe which is connected to STB input of 8282.

Latches are 3 - stated by the AEN signal produced by the DMA controller in the case of the DMA cycle. Data are buffered by th means of two 8286 3 - state octal bidirectional buffers. Direction of transmission is switched by the WR signal which determines the data path. Output of buffers are enabled with the REQN low active signal, which signalizes the data transmission in progress.

3. Clock circuits.

The clock circuit ic composed of 74S124 and two 7473 chips. Clock is stabilised by 25 MHz crystal. 74S124 provides two independent clock frequencies. 25 MHz is used for Intel 8202 dynamic RAM controller, which is the maximum clock frequency for this device. The same clock source is used for the CPU 16/10 chip.

25 MHz is divided by 8 by the means of 3 J-K flip-flops to achieve exact synchronization of the CPU cycles and dynamic RAM responses. This can be very important in some real time applications, where the time is critical and the wait states of CPU have to defined. This will be not the case of independent dynamic RAM controller clock. Resulting 3.125 MHz frequency is applied to CPU CLK input. The same frequency is used for USART CLK input and DMA CLK inputs.

Independent 614.4 kHz from the other oscilator of 74S124 is used for USART TxC and RxC inputs for definition of receiver and transmitter baud rates.

4. Input/Output.

4.1. Serial I/O.

The serial interface for the card is based on the Intel 8251A USART chip and provides V.24 interface. The 8251A is fully programable for synchronous or asynchronous operation and enables full or half duplex operation with the peripheral. The receive/transmitt baud rate is supplied by a 74S124 clock generator.

For a testing purposes the chosen frequency 614.4 kHz for TxC and RxC inputs enables 9600 Baud rate communication with thw CRT terminal in asnchronous mode. Fraction of this rate are software programable. For some application it will be necessary to use the programable baud rate generator with the Intel 8253 or other similar devices. List of USART commands can be found in [6].

Selection of 8251A chip is provided by the means of 8205 adress decoder. Adress 30H is used for data I/O and 31H for command and status words. Fig. 4. ilustrates the scheme of serial I/O designed for use with SPC 16/10 CPU chip.

4.2. Parallel I/O.

Parallel I/O system is based on two Intel 8255 chips. There are 32 general purpose I/O lines available on each chip that are divided into three groups of eight. Each group can be initialized in either input or output mode. The programming rules for these I/O lines are those of the 8255 chip. The list of initialization and operation commands of the SPC 16/10 procesor for 8255 device can be found in [6] .

8205 address decoder is used to select two 8255 devices at adresses stated in table:

Address	Port name
10H	Port A 8255 nr 1
11H	Port B 8255 nr 1
12H	Port C 8255 nr 1
13H	Control 8255 nr 1
20H	Port A 8255 nr 2
21H	Port B 8255 nr 2
22H	Port C 8255 nr 2
23H	Control 8255 nr 2

5. Interrupt control.

Interrupt control is acomplished by the two SPC 16/11 Signetics chips. SPC 16/11 is a bipolar LSI device designed to facilitate the handling of multilevel priority interrupts in microcomputer systems. It can work in three modes of operation from which DINXT and INTRA mode is used in designed microcomputer. CINXT mode can be used after the slight modification of the SPC 16/11 nr2 chip connection.

In the DINXT - discrete interrupt transmitter mode, the interrupt handling controller (IHC) can accept eight discrete interrupt request signals (IR0 - IR7) and generates a prioritizied time multiplexed interrupt level code on its bus coded interrupt output BCI. The interrupt priority is determined by the position of the discrete interrupt on the input pins. While transmitting thr BCI, an internal priority resolver checks if a higher priority level interrupt is active from another IHC and causes transmission of its BCI to be terminated if this condition exists.

In the CINXT - coded interrupt transmitter mode, the IHC accepts an interrupt level code on its interrupt level inputs (IL0 - IL5) and serially transmitts the code on its BCI line. Transmission of the BCI takes place as in the DINXT mode.

IHC operating in the INTRA - interrupt receiver and arbitrator mode receives the time multiplexed interrupt codes from the transmitting IHC devices operating in the DINXT and CINXT modes and compares the received code to the code stored in its program level register PLR by the CPU. If the received code has a higher priority level than the PLR the PLR is updated to the received priority level and an interrupt request is generated for the CPU. Upon request from the CPU, the IHC places the interrupt code on its IL0 - IL5 outputs. Communication with the CPU is done through the external register at address FFH, which will be described later in this chapter. Upon request from the CPU the IHC placea the interrupt code on its IL0 - IL5 output lines.

Active low interrupt request inputs IR0N - IR7N from peripheral devices and control units are connected to IHC in DINXT mode. Bus coded interrupt is send on the BCI line by this chip to IHC in INTRA mode. Details of serial coded interrupts can be found in A20.

Interrupt clock is supplied from the clock generator (3.125 MHz) of the CPU. This signal is also applied to the INCL line of the UPL Bus for synchronization of BCI with other interrupt controllers. Bus coded interrupt signal is supplied to the IHC in INTRA mode. If a higher priority interrupt code (lower numeric value) than the current code in the PLR of the

INTRA IHC is received, output flag INTFN is set low.

Active low output INTFN causes that the CPU reads the interrupt level from the external register with address FFH, writes this level into the PL register and external register FFH and vectors to the adress in the interrupt vector table which coresponds to the new program level. ommand strobe CMDSN to the IHC in INTRA mode is produced from the address bus lines with the address FFH and external register control R/W signals. Open colector outputs IL0 - IL5 are directly connected to lowest significant AD input/output lines of the CPU to provide direct reading or writing for register FFH.

6. Memory organization.

For on board memory the firmware memory option was used. 2K of 16-bit words EPROM with starting address 0000H is used for simple monitor program, which was used for testing purposes. Intel 8205 address decoder is used for proper selection of two 2716 EPROMs. 1K of static RAM memory is placed from starting address 1000H. 8205 is again used with the appropriate address line to enable this part of memory. MEMRD and MEMWR control signals are used for enabling writing or reading desired locations of memory. Fig.7. shows the scheme of EPROM and static RAM configuration.

Dynamic RAM management is provided by the means of the Intel 8202 dynamic RAM controller chip. The starting address of dynamic RAM is 8000H with designed firmware memory layout. Sixten Intel 2118 provide 16K of 16 - bit words of RAM memory.

8202 provides refresh cycles in proper intervals and arbitrates in the cases of simultaneous memory accesses attempts. SACK output is used to enter wait states by the CPU if necessary. XACK signal strobes output data from the memory which are enabled on the local data bus by the means of MEMRD ored with PCS signals. PCS signal is produced from FIRMN and A0 lines to access upper 16K of firmware memory.

Firmware memory for practical realization of designed Central Instruction Processor was used because of internal switch of the CPU to firmware state after the reset and power fail interrupt. Timing analysis of the used memories proved that there are no wait states necessary to be inserted exept of the dynamic RAM. One wait state is automatically inserted by the dynamic RAM controller with every dynamic RAM memory access.

7. Bus interface.

Microcomputer bus interface is based on the SPC 16/12 Unified Bus Manager (UBM) chip. This bipolar device is designed to permit a bus master device to request and gain control of the common system bus. SPC 16/12 allows a bus master device in a multimaster systems to communicate with other masters as well as to transfer data to a common memory and I/O devices. UBM is primarily intended to interface devices to the Signetics proposed UPL Bus. Full details of the structure and operation *of this bus can be found in [4]* .

UBM operates in designed scheme in CPU mode, its CU input is tied low. UBM serves to allocate the bus to its own use (highest priority) and also acts as a bus controller in granting bus accesess to other master devices. The CPU requests its own access to the bus by providing the signals BRQA, BRQB and WR which specify the type of bus exchange request along with the bus request strobe *BRQS1*.

Bus request strobe BRQS1 is produced by an of-board access logic multiplied with the REQN sgnal from the CPU, signalizing data transfer in progress. The positive edge of the bus request strobe stores the request type in bus allocation logic.

Signals AMA and AMB from the CPU are directly connected to the BRQA and BRQB inputs of the UBM. The following bus exchange types are specified by means of these signals:

	BRQA	BRQB
memory access with lock	0	0
memory access without lock	0	1
peripheral access	1	0
external register access	1	1

The lock mode access provides the bus master with the capability of retaining control of the bus for more than one access cycle. The UBM allocates the bus usage to the CPU after it has received the bus request strobe BRQS1 (of-board adress detection circuitry is provided for this purpose) and the following conditions are satisfied:

No other master has requested the bus (BUSRN high).

Another master has not been selected (MSN high).

The bus is not being used (BSYN high for a minimum of BSYNDLY period).

The UBM assumes control of the bus by lowering BSYN. Once

the bus has been allocated, the UBM generates the necessary control ans timing signals automatically :

EMADN is used to enable two Intel 8282 latches to transfer address on the UPL Bus. Address is strobed by the ASTRO strobe from the CPU.

EBON signal is used for switching the direction of data flow trough two 8286 buffers and EMADN enables data to a local or to a system bus. Byte swapping is not provided.

Timing signals TMRN, TMPN or TMEN are dependent on the type of bus exchange requestet. Upon a reception of a reply timing signal (TSMN) from the addressed device acknowledging the data transfer, the UBM will raise its RDY line signifying to the CPU that the bus transfer has been completed. Fig.8 shows the UPL Bus interface tu the single board computer with the SPC 16/10 CPU. Not shown are the drivers for the control signals. Specification of these can be found in [4].

8. DMA control.

DMA control is provided using the Intel 8237 device. It is a 4 - channel programable DMA controller. There are four pairs of channel registers, each pair consisting of a 16 - bit DMA address register and a 16 - bit terminal count register. 8237 also includes one 8 - bit status register. By the means of the 8205 adress decoder the following address assignement is used :

I/O port address	Function	F/L
00H	CH0 DMA address LSB	0
00H	CH0 DMA address MSB	1
01H	CH0 Term. count LSB	0
01H	CH0 Term. count MSB	1
02H	CH1 DMA address LSB	0
02H	CH1 DMA address MSB	1
03H	CH1 Term. count LSB	0
03H	CH1 Term. count MSB	1
04H	CH2 DMA address LSB	0
04H	CH2 DMA address MSB	1
05H	CH2 Term. count LSB	0
05H	CH2 Term. count MSB	1
06H	CH3 DMA address LSB	0
06H	CH3 DMA address MSB	1
07H	CH3 Term. count LSB	0
07H	CH3 Term. count MSB	1
08H	Mode (Wr)	0
08H	Status (Rd)	1

Upper part of the address is latched by the means of Intel 8282 at the ADSTB strobe which is produced by controller. AEN signal is uded to switch the direction of transmission of 8286 buffer for input or output the control sgnals. signal is also used for disabling the CPU control signals buffer DM8097. It is also used for 3 - stating 8282 address latches used by CPU for address demultiplexing.

DMA request and DMA acknowledge signals are connected through the inverters to the CPU appropriate pins. Fig. 9. shows complete scheme of DMA logic for designed microcomputer.

Literature.

[1] SPC 16/10 Signetics product data sheet

[2] SPC 16/11 Signetics product data sheet

[3] SPC 16/12 Signetics product data sheet

[4] UPL - Bus Specification, Philips Data Systems 1979

[5] Intel Data Catalog 1980 , Intel Corp.

[6] SBCA Single Board Computer, Philips Data Systems 1980

[7] P800M Programmers Guide 1 & 2, Volume II: Instruction Set, Philips Data Systems 1976.

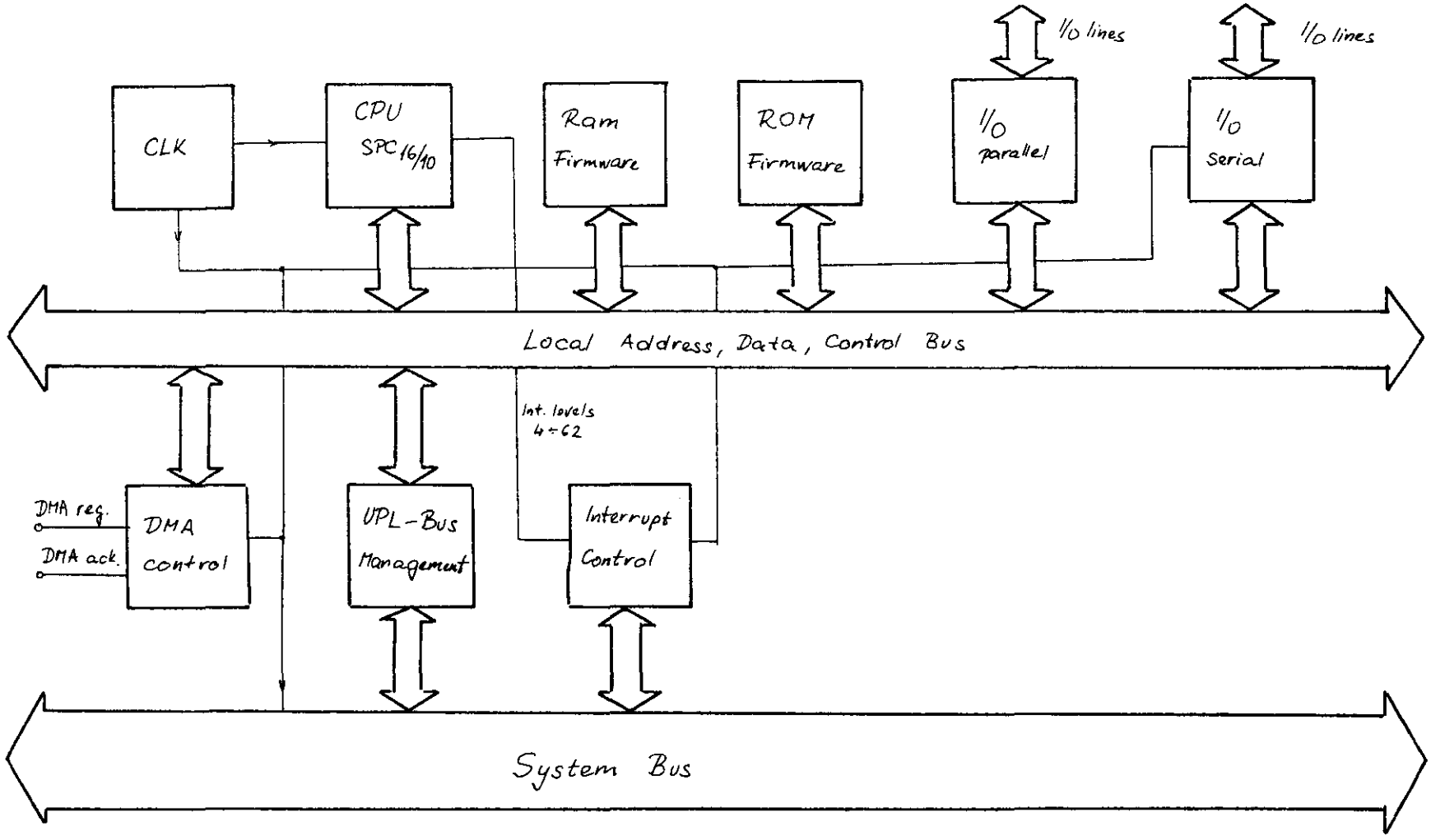

Fig. 1. Block diagram of the Central Instruction Processor.

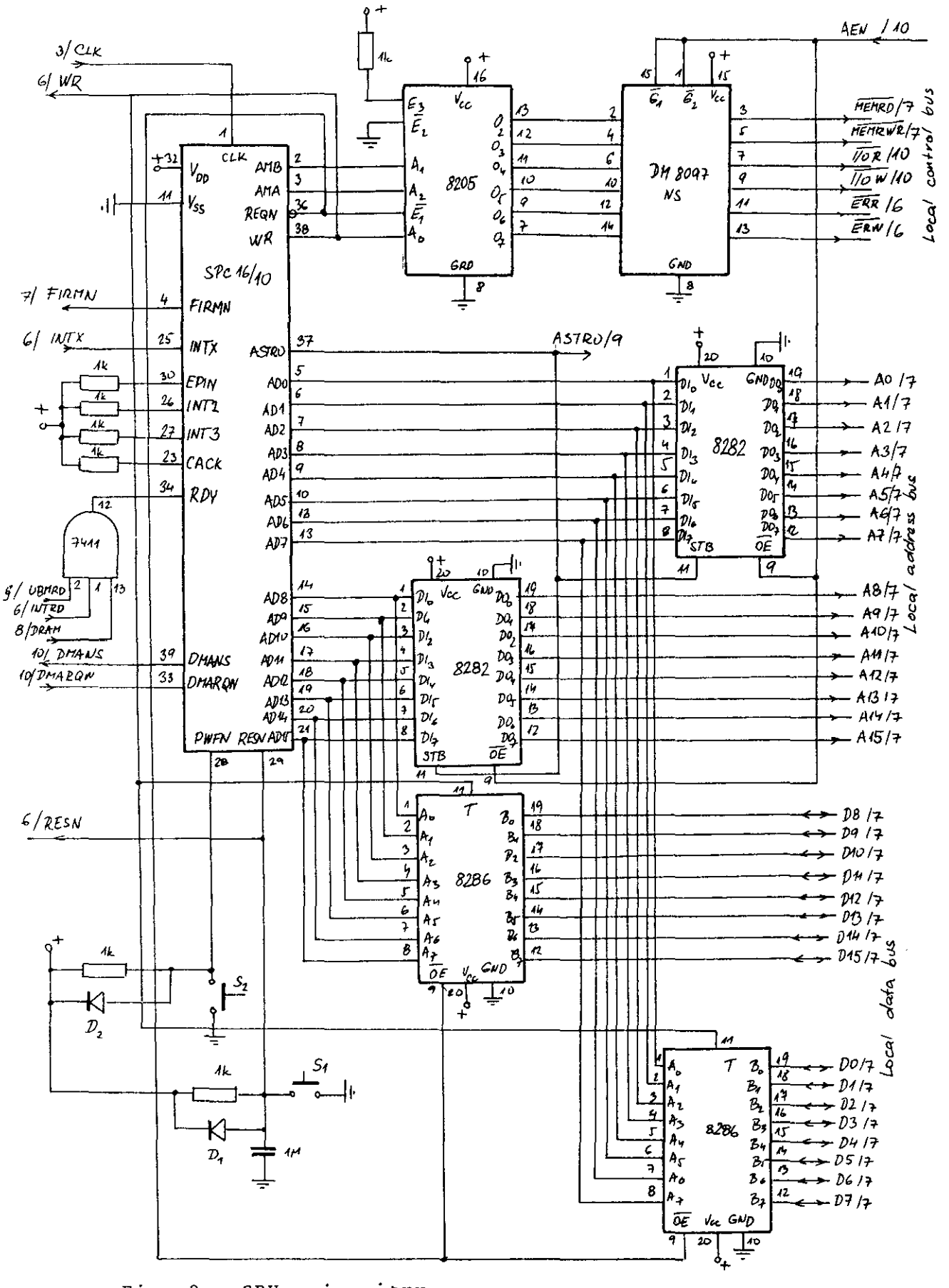

Fig. 2. CPU circuitry

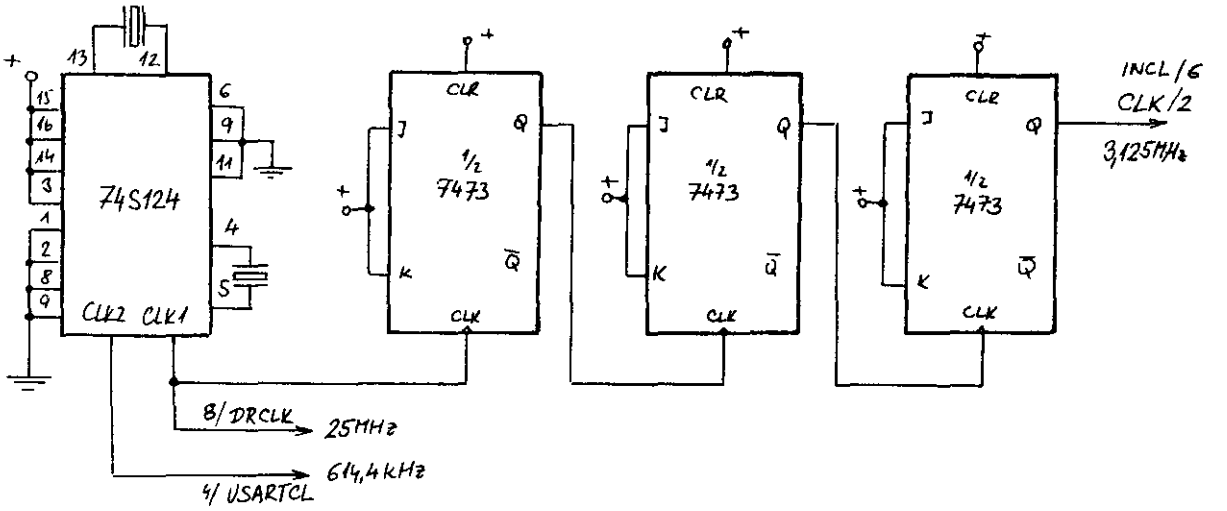

Fig. 3. Clock circuits

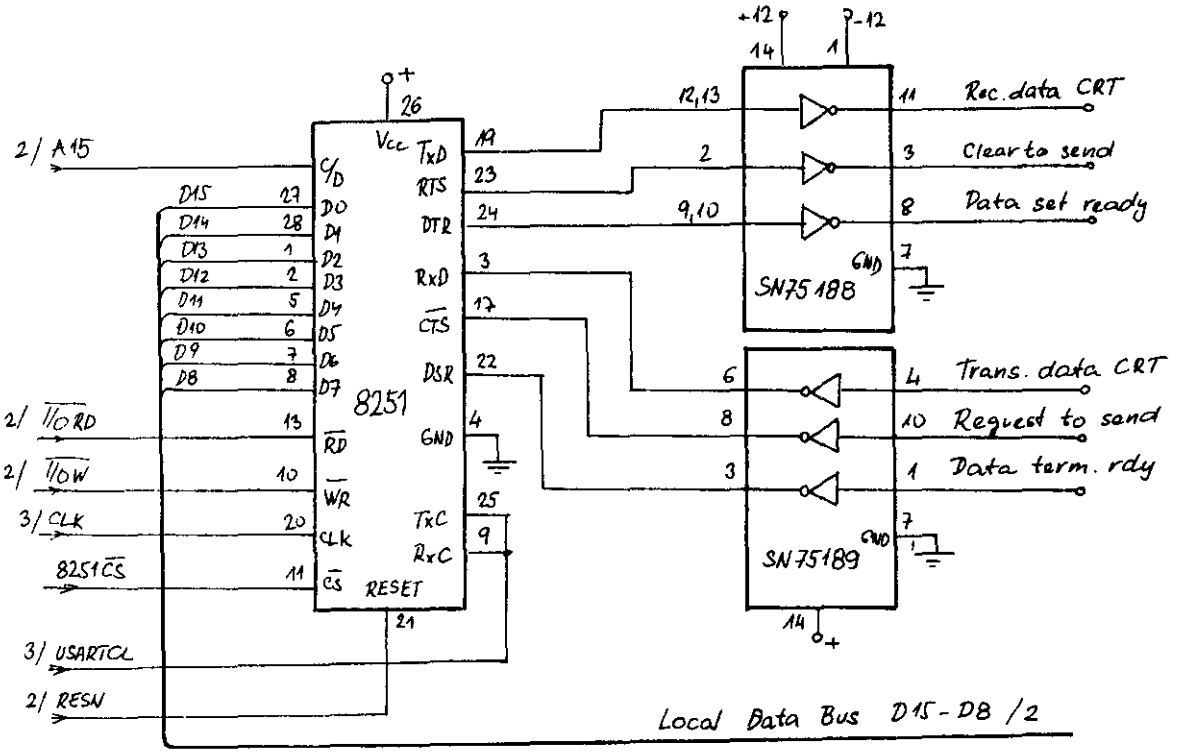

Fig. 4. Serial Input/Output

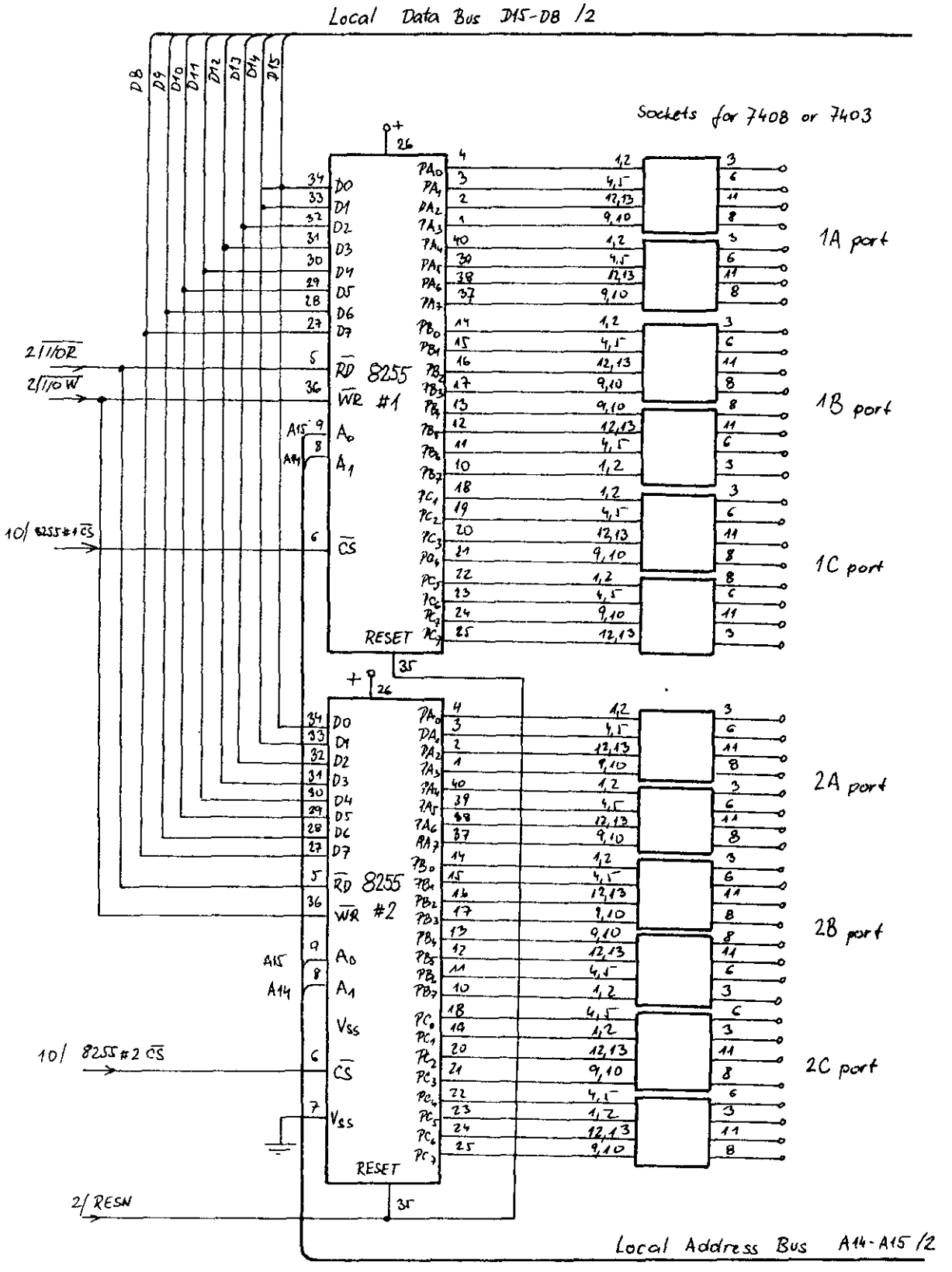

Fig. 5. Parallel Input/Output

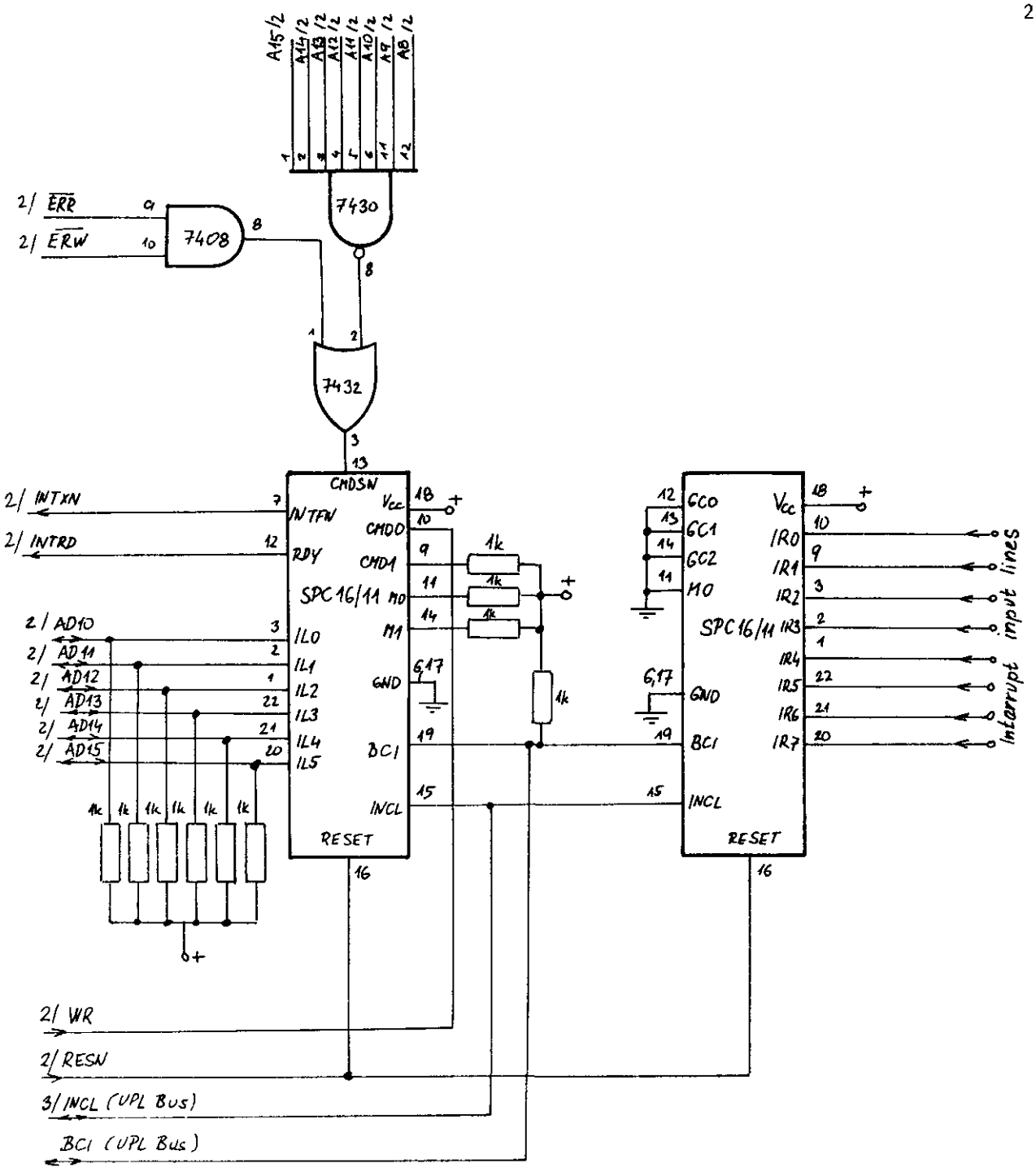

Fig. 6. Interrupt control

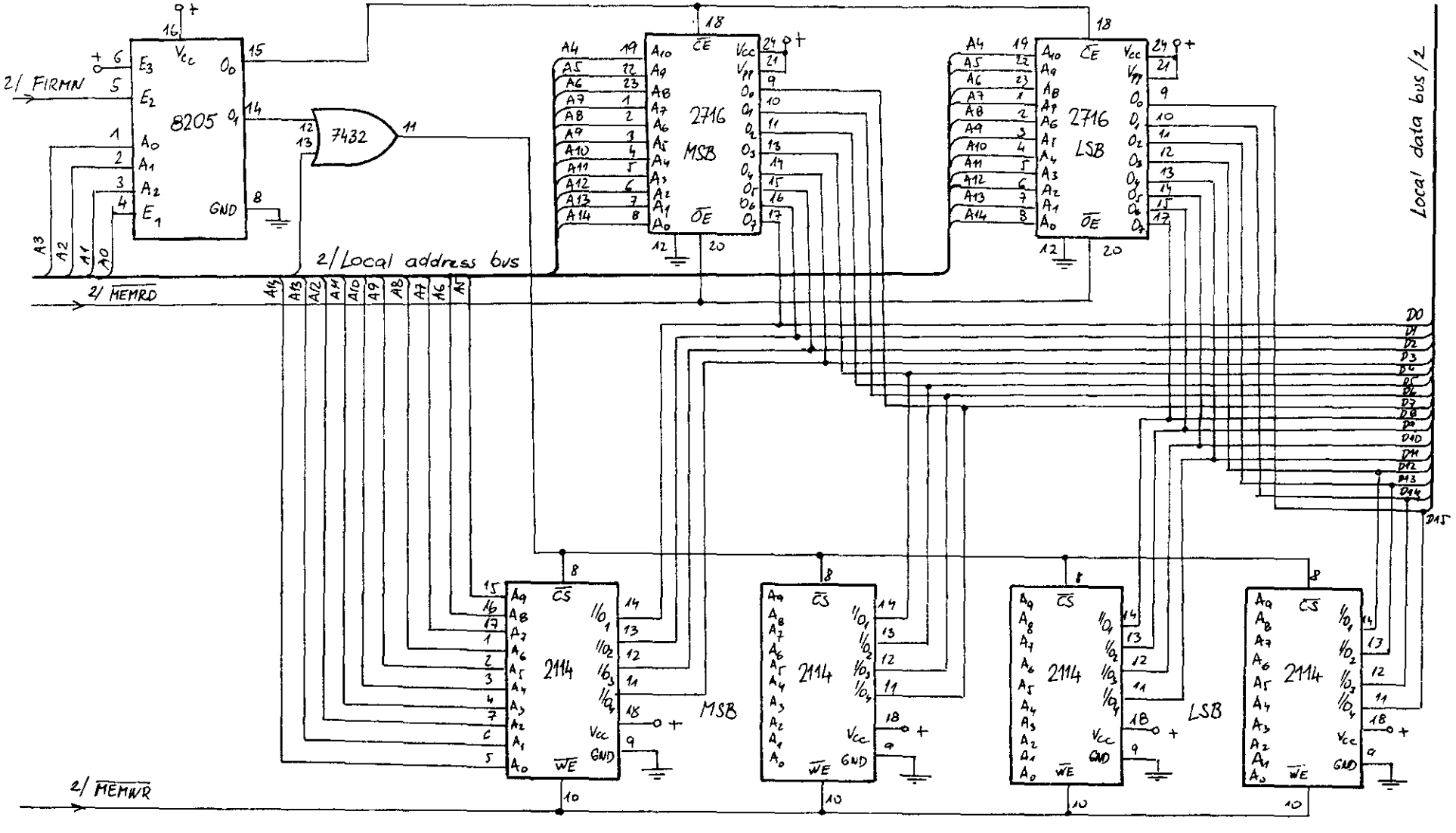

Fig. 7. PROM and static RAM diagram

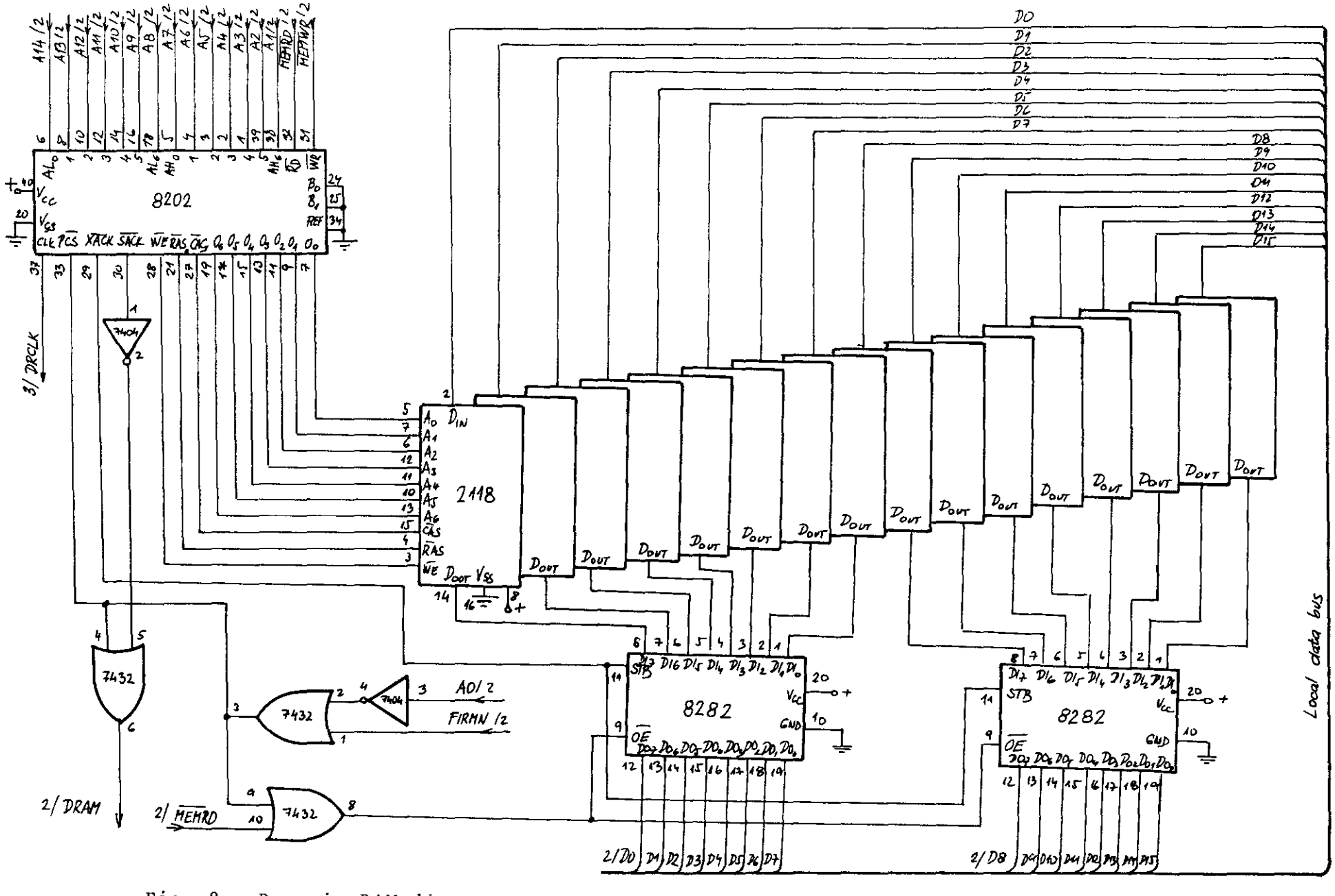

Fig. 8. Dynamic RAM diagram

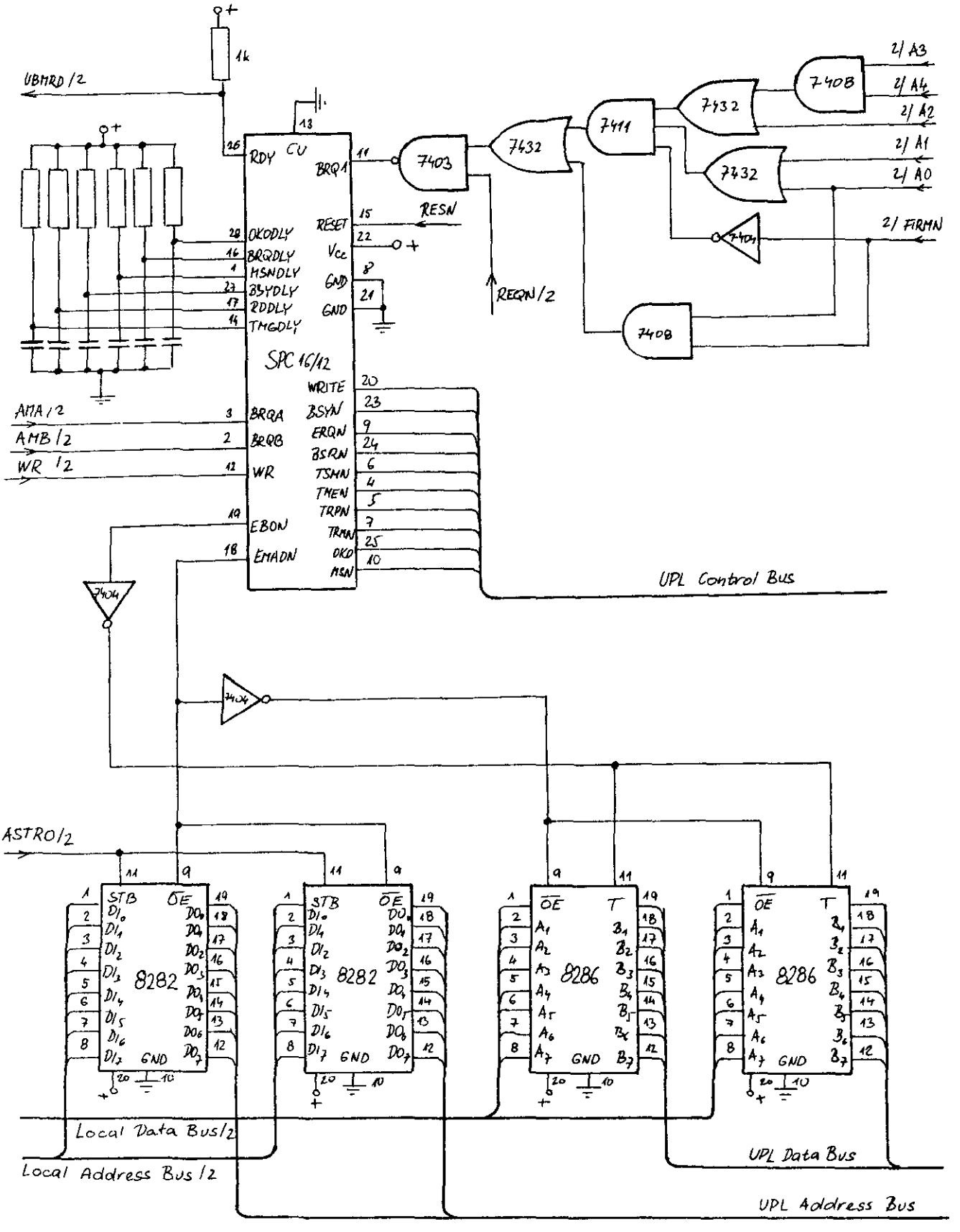

Fig. 9. Multimaster Bus Management

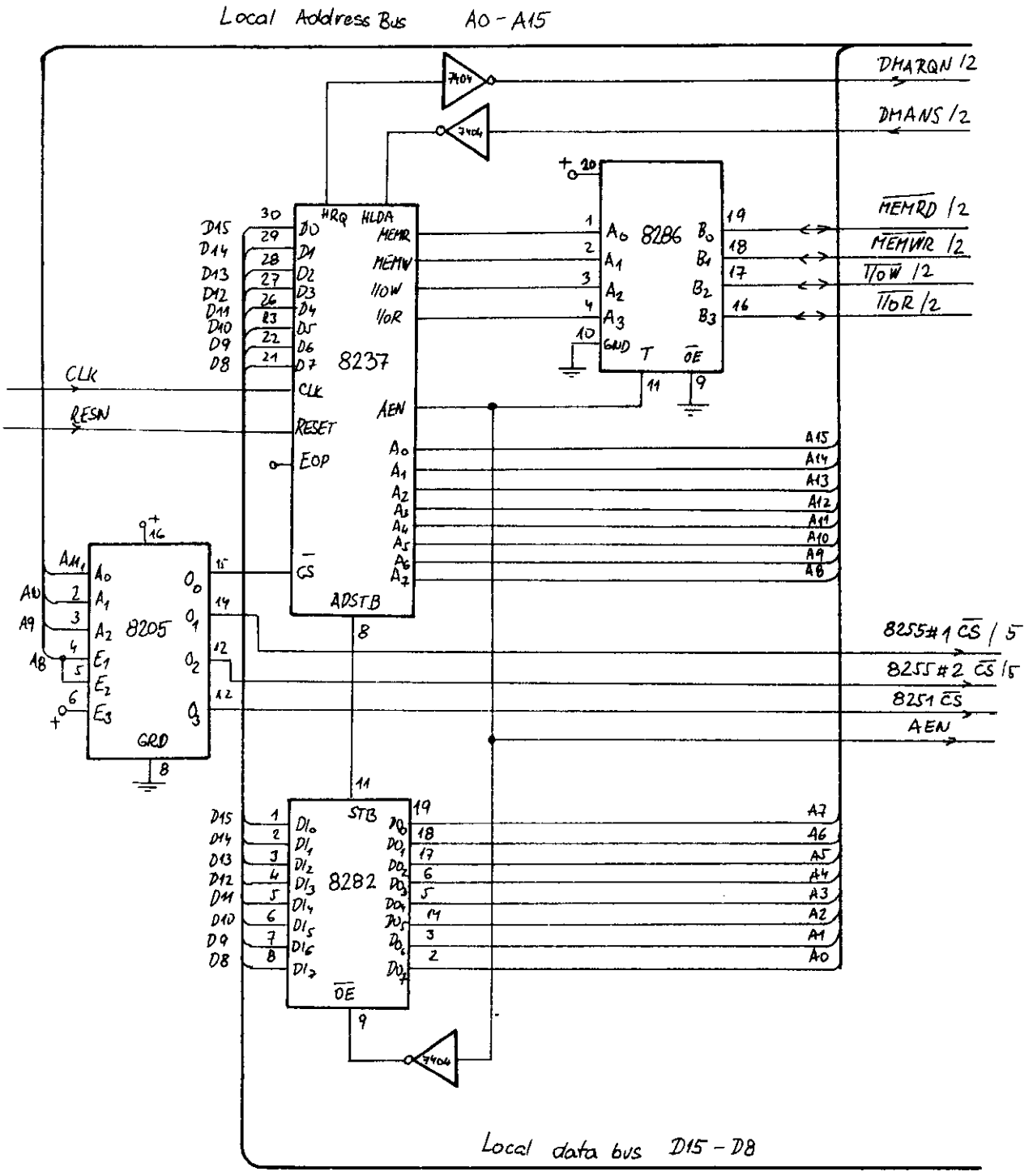

Fig. 10. DMA control

EINDHOVEN UNIVERSITY OF TECHNOLOGY
THE NETHERLANDS
DEPARTMENT OF ELECTRICAL ENGINEERING

Reports:

105) Videc, M.F.
STRALINGSVERSCHIJNSELEN IN PLASMA'S EN BEWEGENDE MEDIA: Een geometrisch-optische en een golfzonebenadering.
TH-Report 80-E-105. 1980. ISBN 90-6144-105-6

106) Hajdasiński, A.K.
LINEAR MULTIVARIABLE SYSTEMS: Preliminary problems in mathematical description, modelling and identification.
TH-Report 80-E-106. 1980. ISBN 90-6144-106-4

107) Heuvel, W.M.C. van den
CURRENT CHOPPING IN SF_6.
TH-Report 80-E-107. 1980. ISBN 90-6144-107-2

108) Etten, W.C. van and T.M. Lammers
TRANSMISSION OF FM-MODULATED AUDIOSIGNALS IN THE 87.5 - 108 MHz BROADCAST BAND OVER A FIBER OPTIC SYSTEM.
TH-Report 80-E-108. 1980. ISBN 90-6144-108-0

109) Krause, J.C.
SHORT-CURRENT LIMITERS: Literature survey 1973-1979.
TH-Report 80-E-109. 1980. ISBN 90-6144-109-9

110) Matacz, J.S.
UNTERSUCHUNGEN AN GYRATORFILTERSCHALTUNGEN.
TH-Report 80-E-110. 1980. ISBN 90-6144-110-2

111) Otten, R.H.J.M.
STRUCTURED LAYOUT DESIGN.
TH-Report 80-E-111. 1980. ISBN 90-6144-111-0 (in preparation)

112) Worm, S.C.J.
OPTIMIZATION OF SOME APERTURE ANTENNA PERFORMANCE INDICES WITH AND WITHOUT PATTERN CONSTRAINTS.
TH-Report 80-E-112. 1980. ISBN 90-6144-112-9

113) Theeuwen, J.F.M. en J.A.G. Jess
EEN INTERACTIEF FUNCTIONEEL ONTWERPSYSTEEM VOOR ELEKTRONISCHE SCHAKELINGEN.
TH-Report 80-E-113. 1980. ISBN 90-6144-113-7

114) Lammers, T.M. en J.L. Manders
EEN DIGITAAL AUDIO-DISTRIBUTIESYSTEEM VOOR 31 STEREOKANALEN VIA GLASVEZEL.
TH-Report 80-E-114. 1980. ISBN 90-6144-114-5

115) Vinck, A.J., A.C.M. Oerlemans and T.G.J.A. Martens
TWO APPLICATIONS OF A CLASS OF CONVOLUTIONAL CODES WITH REDUCED DECODER COMPLEXITY.
TH-Report 80-E-115. 1980. ISBN 90-6144-115-3

EINDHOVEN UNIVERSITY OF TECHNOLOGY
THE NETHERLANDS
DEPARTMENT OF ELECTRICAL ENGINEERING

Reports:

116) Versnel, W.
THE CIRCULAR HALL PLATE: Approximation of the geometrical correction factor for small contacts.
TH-Report 81-E-116. 1981. ISBN 90-6144-116-1

117) Fabian, K.
DESIGN AND IMPLEMENTATION OF A CENTRAL INSTRUCTION PROCESSOR WITH A MULTIMASTER BUS INTERFACE.
TH-Report 81-E-117. 1981. ISBN 90-6144-117-X